0300

473

R and D

TARADCOM LABORATORY TECHNICAL REPORT

NO. [redacted] 12242
TSC-PD-A156-1

REPRO-MODELING APPLIED TO THE SIMPLIFICATION OF TARADCOM COMPUTER MODELS

December 1976

Contract DAAE07-76-C-0105

by Gene B. Goldstein
Theodore Dushane
Technology Service Corporation
Santa Monica, California 90403

U.S. ARMY TANK-AUTOMOTIVE
RESEARCH AND DEVELOPMENT COMMAND
Warren, Michigan 48090

ABSTRACT

As mathematical models become large and complex, they become more difficult to apply and understand. Repro-modeling is a technique for constructing an efficient input/output approximation to a complex model. The technique is founded upon an economical representation for continuous piecewise linear functions of many input variables. Repro-modeling as described here is the fitting of input/output data by a continuous piecewise linear function. A software package called COMPLIAR which accomplishes this is described, and the theory underlying its use is explained. Instructions for the use of the software and examples are included.

PREFACE

This document describes an efficient computational procedure for repro-modeling a sophisticated model. The procedure makes use of multivariate linear regression theory to develop a continuous, multivariate, piecewise-linear approximation to the relationship that exists between the input and output variables of the original model. Specially designed software (COMPLIAR) is described for obtaining the repro-model using representative input/output data obtained from the original model. Thus, an effort has been made to provide in one place: an introduction to the repro-modeling concept, an exposure to the mathematical principles involved, a tour through a specific implementation known as COMPLIAR, and an illustrative example.

Though this document is not a User's Manual as such, the reader who is primarily user oriented will find Sections 4 and 5 most pertinent, and the other sections may be omitted.

TABLE OF CONTENTS

Section		Page
I.	INTRODUCTION	1
	1.1 PROBLEM DESCRIPTION AND DESCRIPTION OF EFFORT	1
	1.2 MOTIVATIONS FOR REPRO-MODELING	4
II.	CONTINUOUS MULTIVARIATE PIECEWISE-LINEAR APPROXIMATION/REGRESSION (COMPLIAR)	7
	2.1 COMPLIAR DEFINED	
	2.2 IMPLEMENTING COMPLIAR	11
	2.2.1 Basic Formulas	11
	2.2.2 Mean-Squared Error Optimization Criterion	19
	2.2.3 The Optimization Equations	21
	2.2.4 Implementing the Optimization Procedure	29
III.	DETAILED IMPLEMENTATION DESCRIPTION	31
	3.1 OVERVIEW	31
	3.1.1 Input and Standardization of the Data	32
	3.1.2 Initialization of the P-Functions and Marginal Optimization	33
	3.1.3 Sequential Optimization	35
	3.1.4 Joint Optimization	36
	3.1.5 Removing Inactive Hyperplanes	37
	3.1.6 Returning to Original Coordinates	37
	3.2 ESSENTIAL INGREDIENTS OF THE THREE TYPES OF OPTIMIZATION	39
	3.2.1 P-Function Initialization (Used in Marginal Optimization)	40
	3.2.2 Optimization of the Weights	45
	3.2.3 Gradient Search Scheme	45
	3.3 ALTERNATE MODES OF PROGRAM OPERATION	47
	3.3.1 Calculate Statistics on Input Data	48
	3.3.2 Refine Previously Determined P-Functions	48

TABLE OF CONTENTS (Continued)

Section			Page
IV.	USE OF THE PROGRAM.		52
	4.1 STEP-BY-STEP PROGRAM USE		52
		4.1.1 Initial Data Analysis Alone	52
		4.1.2 Initial Run of the Program.	54
		4.1.3 Starting Off the Optimization Process from Previously Determined P-Functions	54
	4.2 PROGRAM INPUT PARAMETERS		56
	4.3 ERROR MESSAGES AND COMMON PROBLEMS ENCOUNTERED		63
	4.4 PRINTED OUTPUT OF THE PROGRAM.		66
	4.5 OTHER OUTPUT OF THE PROGRAM.		73
V.	SOME EXAMPLES		74
	5.1 A SAMPLE RUN		74
	5.2 AN 8-DIMENSIONAL EXAMPLE		110
VI.	PROGRAM DESCRIPTION		112
	6.1 THE MAIN PROGRAM AND ITS STRUCTURE		113
	6.2 INPUT ROUTINES: SMNPUT AND SMINDAT.		113
	6.3 OPTIMIZATION ROUTINES.		113
		6.3.1 INITA	113
		6.3.2 OPTW.	119
		6.3.3 OPT	119
		6.3.4 The Function Subroutine F and the Subroutine CFCN	119
		6.3.5 STPOUT.	121
		6.3.6 STOPG	121
		6.3.7 Output P-Functions: WSFUNC and WSFUNF. . . .	121
		6.3.8 ENERGY.	121
	6.4 FSTATS		122
		6.4.1 MNCOV	122
		6.4.2 NETHYP.	122
		6.4.3 Statistical Subroutines Called by NETHYP: RGSTAT and ERCALC	123
APPENDIX A.			A-1
DISTRIBUTION LIST			
DD FORM 1473			

LIST OF FIGURES

Figure		Page
1.	The Army Mobility Model.	8
2.	An Example of a Continuous Piecewise-Linear Fit Involving a Single Input Variable.	13
3.	An Example of a Continuous Piecewise-Linear Function in Two Variables	13
4.	A P-Function in One-Dimension.	16
5.	Basic Form of Initializing Function (Optimization Initialized Over the Variable x_k).	42
6.	Initial COMPLIAR Fit For Q = 4	44
7.	Main Program and its Structure	114
8.	Tree Structure of the Program.	115
9.	Commented IFTRAN Listing of Main Program (CMPLAR).	116-117
10.	Commented IFTRAN Listing	118
11.	Commented IFTRAN Listing of OPT.	120

LIST OF TABLES

Table		Page
4.1	DD PARM Namelist (Data Definition).	57
4.2	INPARM Namelist (Input Parameters).	58
4.3	STPARM Namelist (Structure Parameters).	59
4.4	ITPARM Namelist (Iteration Parameters).	60
4.5	OTPARM Namelist (Output Parameters)	61&62
4.6	Error Messages.	64
5.1	Listing of Input Cards.	75
5.2	Repro-Model for 8-Dimensional Case.	111

I. INTRODUCTION

1.1 PROBLEM DESCRIPTION AND DESCRIPTION OF EFFORT

The U.S. Army Tank-Automotive Research and Development Command (TARADCOM) is concerned with improving the design of wheeled and tracked vehicles to achieve better ground mobility and higher rates of survivability. To pursue these objectives, TARADCOM has invested considerable effort in the development of a highly detailed computer model for vehicle ground mobility. This model was first comprehensively documented in [1] which describes the model's status as of July 1971. This first generation model is referred to as the AMC '71 Mobility Model.

The Mobility Model has evolved as various embellishments were added and modifications made in order to ensure that the model authentically accounted for the principal factors affecting vehicle mobility. However, even the 1971 version of the Mobility Model is sufficiently intricate to motivate the use of approximations to the model. Such approximations are highly effective in reducing computer cost and program size and in simplifying the interpretation of model outputs. This is especially important when it is desired to use the Mobility Model as a design tool or as part of another model, e.g., a survivability model. In these situations only a few of the hundreds of parameters which affect vehicle mobility are of immediate interest. Yet, without any model approximation, it is necessary to rerun the entire mobility model in order to investigate the effect of changing the few parameters of interest. Extensive parametric examination of vehicle mobility is impractical in this situation.

---

[1] TACOM Technical Report No. 11789 (LL143), The AMC '71 Mobility Model, July 1973. AD766733, AD766734.

To overcome this problem, TARADCOM is exploring the application of repro-modeling as a means of approximating the Mobility Model.

Repro-modeling [2] is an approach for creating an efficient input/output approximation to a sophisticated model. The repro-modeling approach is based upon obtaining a simple mathematical expression for the desired input/output relationship. It is an empirical approach which employs the original model to supply the requisite data (typically tables of model output values for a selected range of input values). Once the input/output data are at hand, the idea is to approximate the functional relationship implicitly imbedded in the data with a mathematical expression which is both economical to use and which reasonably approximates the behavior of the data.

To demonstrate the potential utility of repro-modeling in approximating the Mobility Model, TSC, under contract to TARADCOM, has developed a computer program which is capable of repro-modeling a model when the input may contain as many as eight input variables. The computer program produces a multi-dimensional functional approximation which relates the selected input variables to the output variable (the average velocity for given terrain conditions and a given vehicle for the Mobility Model). Once at hand, the resulting functional approximation may be used as a design tool to further study the effects of changes in the chosen input variables upon mobility.

The purpose of this document is to describe the approach taken in obtaining the repro-model and to completely describe the computer software, including instructions for its use.

---

[2] Meisel, W. S., and D. C. Collins, "Repro-Modeling: An Approach to Efficient Model Utilization and Interpretation," IEEE Transactions on Systems, Man, and Cybernetics; Vol. SMC-3, No. 4, July 1973.

Organizationally we first proceed to describe the concepts embedded in the COntinuous Multivariate Piecewise-LInear Approximation/Regression (COMPLIAR) program.* This is the mathematical approach which has been utilized in obtaining the required multi-dimensional approximation. Next we discuss the computer software implementation used in COMPLIAR, including a step-by-step discussion of its use in furnishing a repro-model of the Mobility Model. A sample deck and the resultant computer output are provided to illustrate the operation of the software. (Complete annotated listings of the computer program can be obtained from the Science and Technology Division of the Tank Automotive Research and Development Laboratory.)

Together with the actual computer program being furnished under this contract, this document should enable TARADCOM personnel to exercise the software and realize the cost savings and interpretative advantages of repro-modeling. In particular, the software may be used to repro-model the relationship between other input/output data sets involving up to 8-dimensional input data and a single output variable.

The software was used on sample data from the Mobility Model as a demonstration of its use. One such demonstration utilized a data set consisting of 5184 input/output data points describing the mobility of an M60 Tank as a function of 8 input variables (4 input variables described terrain properties and the other 4 described vehicle design parameters). The repro-model for this test case accounted for more than 94% of the functional variation contained in the data and required approximately 60 CPU seconds of execution on a CDC 6400 to completely specify the repro-model. The repro-model, itself,

*While one should, in general, distinguish an algorithm (e.g., continuous piecewise linear regression) from a specific software implementation of that algorithm (e.g., the COMPLIAR software package), we will, for the sake of brevity, not be precise in making this distinction.

can easily be implemented on a hand calculator (See Section 5.2 page 110) to furnish small volumes of input/output data.

1.2 MOTIVATIONS FOR REPRO-MODELING

It is worthwhile to review here some of the salient factors which motivate the use of repro-models.

Complicated and intricate models that require excessive running time and/or computer storage may be replaced by far more efficient "repro-models" that are largely equivalent for certain purposes to the original models. Repro-models are, effectively, simpler models of such complex models. Repro-models make practicable many model applications that, heretofore, would not have been considered because of exceedingly large requirements on computational resources.

Typically, a great deal of thought and effort is expended in the development of a complex model. However, when the model is to be used operationally, it often has to be greatly simplified or used sparingly because of excessive requirements for computer time or storage. The net result is that much that the model has to offer is sacrificed--and the cost/benefit ratio of model usage appears to be unattractive.

Repro-modeling is often an alternative to this situation. The result is that the full power of the sophisticated model may be brought to bear on analysis of complex problems, and realistic solutions may be obtained. For example, complex systems analyses may now incorporate fully realistic models, yet require minimal computer resources.

In the case of field deployment of sophisticated algorithms for signal or data processing, the use of repro-models may allow these algorithms to be employed in real-time and within very limited storage constraints (e.g., within microprocessors) without sacrificing performance.

Furthermore, analysis of the repro-model input/output data yields an understanding of what the model implies, and consequently provides a means for model verification or improvement.

For the Army Mobility Model, the major objective of repro-modeling is to accomplish a reduction of model running time and program size. This is a typical objective of repro-modeling. For example, a photochemical air pollution repro-model developed by TSC for the Environmental Protection Agency [3] was so simple that the repro-model output could be calculated on a hand calculator in a minute or so; while the original model took one-half hour, using a large general purpose computer, to produce the same results. Also, the simple form of the photochemical repro-model greatly facilitated an understanding of the degree of reduction in pollutant emissions required in order to achieve a given level of air quality, as implied by the original complex model. It also revealed weaknessess in the original complex model, particularly a large dependence of the output on assumed boundary conditions.

It is often not obvious that repro-modeling is applicable in a given application without detailed examination. For example, some less obvious uses of repro-modeling analyses include the following:

---

[3] Horowitz, Alan, W. S. Meisel, and D. C. Collins, "The Application of Repro-Modeling to the Analysis of a Photochemical Air Pollution Model," EPA Report No. EPA-650/4-74-001, December 1973.

- Determination of which variables most affect the output: the degree of error incurred in ignoring (or not having measurements of) the less important variables.
- Avoiding on-line optimization: a model involving an optimization process can be run off-line, and the relationship of input variables to optimum values of parameters can be repro-modeled.
- Repro-modeling differential equations: a time series of values generated by a model can be analyzed to obtain a simple difference equation, perhaps with a larger time step, which closely approximates the behavior of the original model.

It is hoped that the above discussion will stimulate the imagination of those who will have an opportunity to apply the software described in this document to repro-modeling the Mobility Model and other complex models used by the Army.

II. CONTINUOUS MULTIVARIATE PIECEWISE-LINEAR APPROXIMATION/REGRESSION (COMPLIAR)

As stated in the Introduction, the objective of repro-modeling is to furnish an approximate functional relationship between the inputs and outputs of a complex system so that the repro-model is more efficient to run and easier to interpret than the original model.

One means of achieving this objective, insofar as the Mobility Model is concerned, is to make use of a technique which we will refer to as Continuous Multivariate Piecewise-Linear Approximation/Regression (COMPLIAR).

2.1 COMPLIAR DEFINED

Mathematically, COMPLIAR is a technique for creating a functional form which approximates the multivariate relationship between a set of input variables and one or more output variables.

Thus, consider the system illustrated in Figure 1. For the case at hand the "system" constitutes the Mobility Model. The inputs consist of a set of variables such as vehicle design parameters and terrain descriptive parameters, and the output consists of a measure of the ground mobility of the vehicle such as the maximum velocity attainable under a given set of terrain conditions.

It is convenient to designate the input variables $x_1, x_2, \ldots, x_n$ utilizing vector notation; so that

$$\underline{x} = (x_1 \ldots x_k \ldots x_n) \qquad (2\text{-}1)$$

describes the vector input to the system. In the present application in which the output is the single quantity, speed, it is convenient to designate the generic output variable with the scalar y.

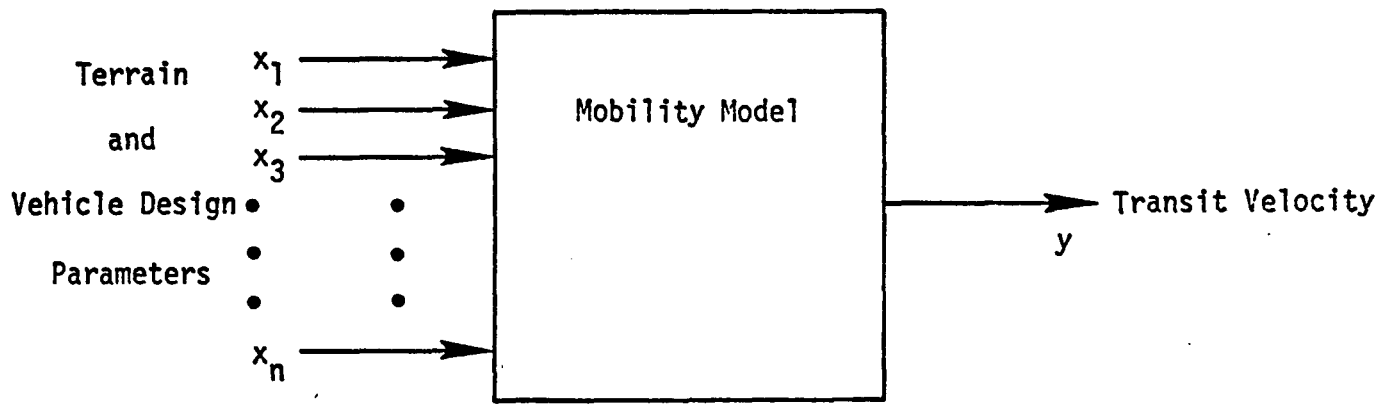


Figure 1. The Army Mobility Model

Underlying our entire procedure is the fact that there exists a relationship between y and $\underline{x}$ which we may summarize by writing

$$y = f(\underline{x}) \tag{2-2}$$

That is, there exists a rule for mapping an n-dimensional point in space determined by the components of $\underline{x}$ into a point (one-dimensional space) determined by the value of y. The objective of repro-modeling may be mathematically stated as arriving at an efficiently executed and accurate approximation to the function $f(\cdot)$ using sets of values of $\underline{x}$ and the concomitant variable, y.

We shall denote the $\ell^{\underline{th}}$ vector data point

$$\underline{z}^{\ell} = (\underline{x}^{\ell}, y^{\ell}) \; ; \; \ell = 1,2,\ldots,M \tag{2-3}$$

and the set

$$\mathcal{Z} = \left\{\underline{z}^{1} \cdots \underline{z}^{M}\right\} \tag{2-4}$$

will be used to designate the set of M observations upon which the repro-model is to be based.

Now, there are many mathematical techniques for approximating $f(\cdot)$ given the data $\mathcal{Z}$. For example, one can, in principle, use an n-dimensional Taylor's series which would lead to a polynomial approximation. The technique of COMPLIAR, which we are about to elaborate upon, has certain features which render it more attractive to the problem at hand than, for example, polynomial approximations. Among its advantages are: ease of interpretation,

better extrapolation behavior, and a greater degree of parsimony than polynomial expansions for approximating functions with regions of sharp curvature.

Before embarking upon the mathematical details of COMPLIAR, let us remark on the properties of the technique which are suggested by its name alone.

The *continuous* aspect of COMPLIAR refers to the fact that the sought for approximation will be a function which is continuous in all of its variables. This is not always a requirement of repro-modeling. Indeed, there are systems which exhibit discontinuous behavior, at least macroscopically. However, for the application of repro-modeling to the Mobility Model, especially with design study applications in mind in which one typically wishes to examine effects of relatively small changes in the input variables, the continuity requirement seems eminently reasonable.

The *piecewise-linear* aspect of COMPLIAR describes the underlying functional form which is used in order to compose the final functional approximation.

A piecewise-linear function is one which is linear in subregions:

$$y = \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + a_{1,n+1} & \text{for } \underline{x} \in X_1 \\ \vdots & \\ a_{R1}x_1 + a_{R2}x_2 + \cdots + a_{Rn}x_n + a_{R,n+1} & \text{for } \underline{x} \in X_R \end{cases} \qquad (2\text{-}5)$$

where $X_1, X_2, \ldots, X_R$ are disjoint subregions covering the region of interest for the input variables.* Each line of Equation (2-5) is specified by

*$\underline{x} \in X_1$ is the conventional mathematical shorthand for "the vector $\underline{x}$ is in the set X_1."

$(n + 1)$ constants: the n coefficients of $x_1, x_2, \ldots x_n$ plus a constant term. Once these constants are specified, each line of Equation (2-5) defines an n-dimensional hyperplane* in the $(n + 1)$-dimensional space spanned by the data points, $\underline{z}^{\ell}$. For this reason we refer to the constants in Equation (2-5) as *hyperplane coefficients*. The determination of these coefficients is commonly referred to as a *regression* problem.

The idea behind COMPLIAR is to suitably obtain the disjoint subregions and hyperplane coefficients so that a satisfactory approximation to the data can be obtained within the framework of the piecewise-linear component structure. An example of such a fit in the one-dimensional case ($y = f(x_1)$) is illustrated in Figure 2. Figure 3 is an example of a fit involving two input variables ($n = 2$).

The remainder of this section is devoted to the discussion of an efficient computational algorithm for implementing COMPLIAR. The algorithm yields a continuous multivariate piecewise-linear function over general (implicitly determined) subregions using a strategy which minimizes the mean squared error in the final approximation.

2.2 IMPLEMENTING COMPLIAR

2.2.1 Basic Formulas

Let x_k denote the $k^{\underline{th}}$ component of the vector, $\underline{x}$, of input variables. The vector $\underline{x}$ defines a point in n-dimensional space for each prescribed value of the input variables. For notational convenience, it will prove to be convenient to augment the $\underline{x}$ vector with an additional element of unity.

*In the two-dimensional case ($n = 2$) a hyperplane reduces to the conventional notion of a plane. In one dimension a straight line results.

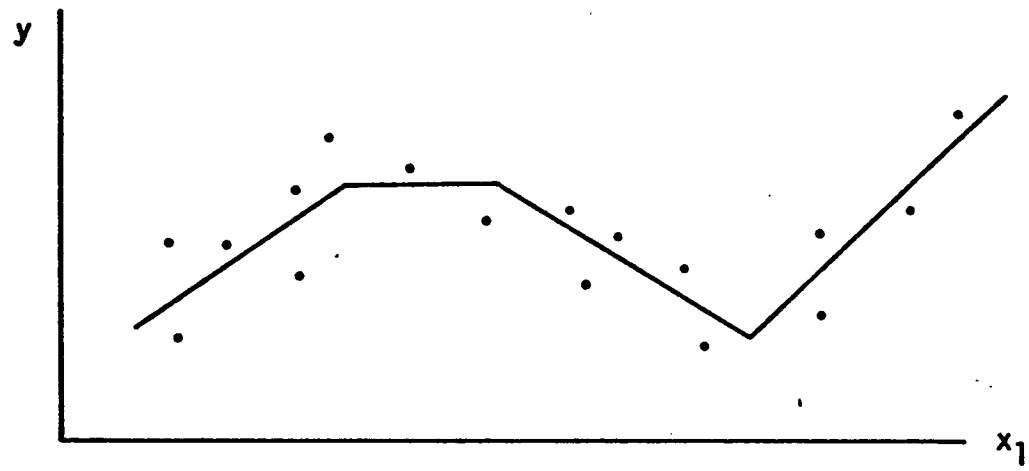


Figure 2. An example of a Continuous Piecewise-Linear Fit Involving a Single Input Variable

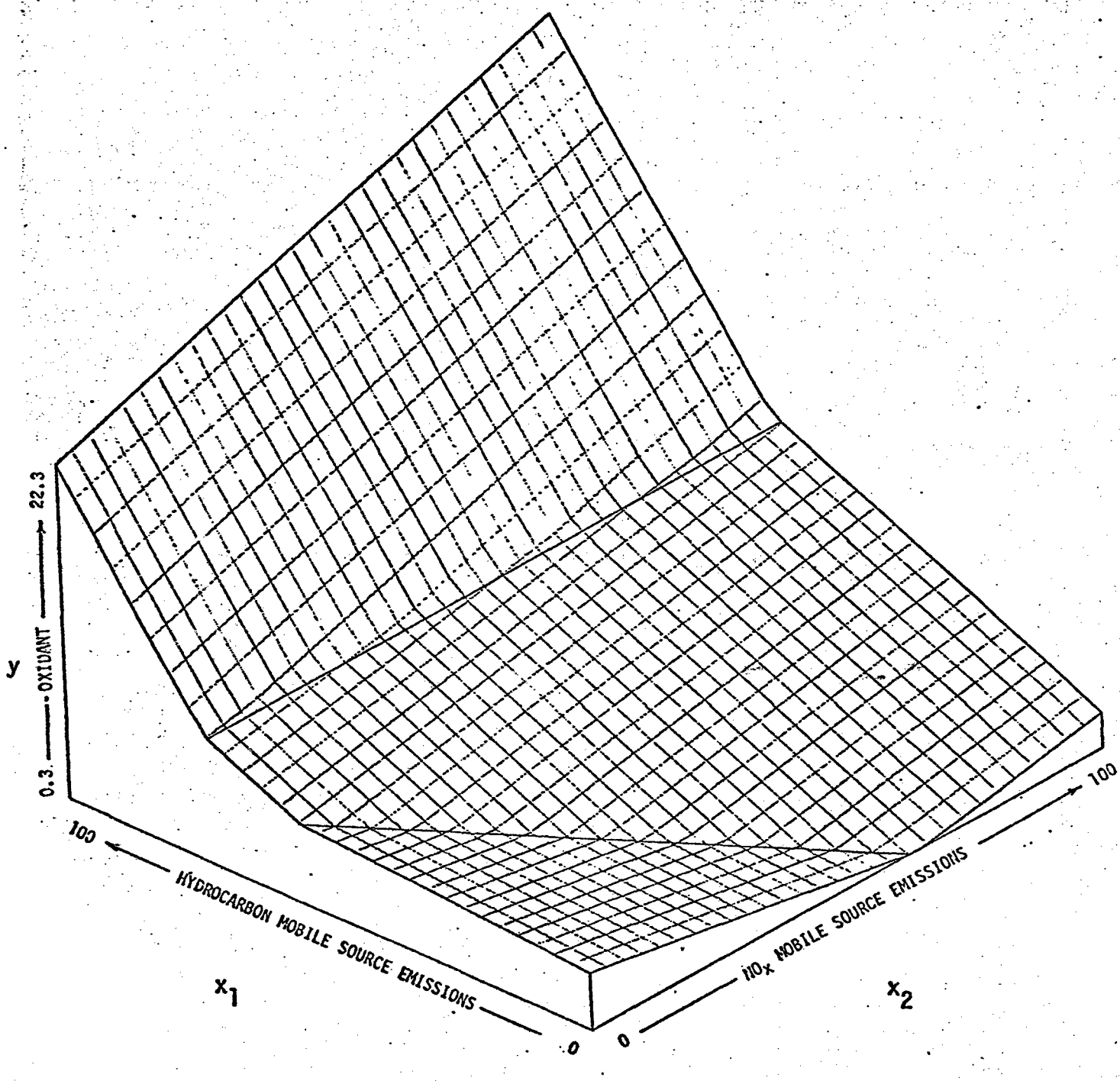


Figure 3. An Example of a Continuous Piecewise Linear Function in Two Variables

Thus, let

$$\underline{x}' = [x_1 \cdots x_k \cdots x_n \; 1] \tag{2-6}$$

Now consider the vector of (n + 1) parameters

$$\underline{a}_q = [a_{q1} \; a_{q2} \cdots a_{qk} \cdots a_{q,(n+1)}] \tag{2-7}$$

Then the function*

$$H_q(\underline{x}) = \underline{a}_q \cdot \underline{x}'^T = \sum_{k=1}^{n} a_{qk} \, x_k + a_{q,(n+1)} \tag{2-8}$$

defines an n-dimensional hyperplane, and $\underline{a}_q$ comprises the vector of hyperplane coefficients.

We wish to compose a function from hyperplanes of the form given by Equation (2-8), in such a way that the composed function is continuous in $\underline{x}$. Thus, suppose we set up a collection of Q hyperplanes via the set of coefficients $\{\underline{a}_q; \; q = 1, 2, \ldots, Q\}$. These coefficients may be conveniently arranged into the Q x (n + 1) matrix shown below.

$$\underline{A} = \begin{matrix} \\ Q \end{matrix} \overset{\overset{\leftarrow (n+1) \rightarrow}{}}{\begin{bmatrix} \underline{a}_1 \\ \vdots \\ \underline{a}_q \\ \vdots \\ \underline{a}_Q \end{bmatrix}} \tag{2-9}$$

*Superscript T above a vector or matrix will be used to denote its transpose.

Then, *define* the function

$$P(\underline{x}) = \underset{1\le q\le Q}{\text{Max}} \{H_q(\underline{x})\} = P(\underline{x};\underline{A}) \qquad (2\text{-}10)$$

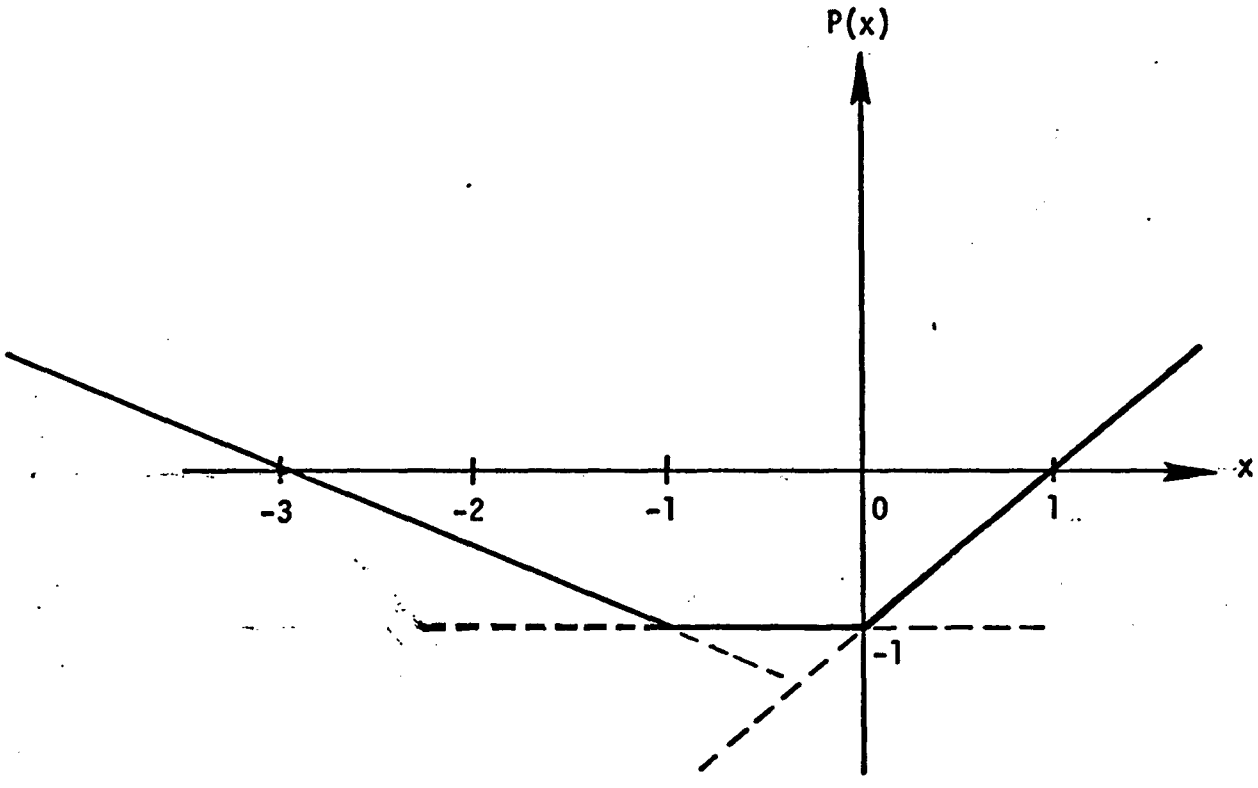
The function P(x) is called a *P-function.* It has the desired continuity property and shall serve as a fundamental building block for arriving at the final functional approximation. The fact that the P-function depends implicitly upon the entire set of hyperplane coefficients has been emphasized via the notation $P(x;\underline{A})$ on the far right of Equation (2-10). Ultimately we will describe a procedure for adjusting $\underline{A}$ to accomplish the desired functional approximation.

Notice in Equation (2-10) that the maximization operation causes one to "switch" from one hyperplane to another in a continuous fashion, depending upon which hyperplane produces the largest value of $H_q(\underline{x})$ for any point $\underline{x}$. In between these switches one remains on a given hyperplane. The ability to switch from one hyperplane to another permits us to produce basic functional building blocks with $P(\underline{x})$ that are highly nonlinear and continuous; even though the constituents are simple linear functions of $\underline{x}$. An example of a P-function is illustrated in Figure 4. This is a one-dimensional example composed of 3 hyperplanes. The hyperplane coefficients are specified by

$$\underline{a}_1 = [-.5 \quad -1.5]$$

$$\underline{a}_2 = [0 \quad -1]$$

$$\underline{a}_3 = [1 \quad -1]$$



Note: Solid curve depicts the P-function which results from applying Equation (2-10) to the 3 lines illustrated above. The dashed extensions show the continuation of the component line segments.

Figure 4. A P-function in One Dimension

and the P-function is determined from

$$P(x) = \text{Max} \left\{ [-.5x - 1.5] \; ; \; -1; [x-1] \right\}$$

However, there is one important limitation to the form of $P(\underline{x})$ and that is that it always produces a convex function.* (This is a consequence of the maximization operation, which happens to be an easy computational method of achieving the continuity.)

Not all functions that we shall choose to approximate are convex. We therefore wish to modify our approach so that we may approximate an arbitrary continuous function while still preserving the simplicity of the functional building blocks described above.

We may remove the convexity limitation by considering the following generic form for the approximating function:

$$F(\underline{x}) = \sum_{i=1}^{N_p} w_i \, P_i(\underline{x}) + w_{N_p+1} \qquad (2\text{-}11)$$

Thus, we simply weight the sum of N_p P-functions using the weights $\{w_i; \; i = 1, \ldots, N_p+1\}$. The use of both positive and negative weights overcomes the convexity limitation.

Our problem is now to take the data, $\mathcal{X}$, and fit it with $F(\underline{x})$. To perform the fit we select a value for N_p, the number of P-functions which our fit will consist of. We also specify the number of hyperplanes which will comprise each P-function. Extending the notation employed in Equation (2-9),

*A convex function is "bowl-shaped".

we let Q_i denote the number of hyperplanes in the $i^{\underline{th}}$ P-function. The hyperplane coefficients for the $i^{\underline{th}}$ P-function are summarized by the matrix $\underline{A}_i$, which has dimension Q_i x (n + 1). The rows of $\underline{A}_i$ are denoted by $\{\underline{a}_q^{(i)}; q = 1, 2,\ldots Q_i\}$. The vector $\underline{a}_q^{(i)}$ comprises the coefficients of the $q^{\underline{th}}$ hyperplane in the $i^{\underline{th}}$ P-function. We then have, for the $i^{\underline{th}}$ P-function (using a generalized form of Equation (2-10)):

$$P_i(\underline{x}) = \underset{1\le q\le Q_i}{\text{Max}} \{H_q^{(i)}(\underline{x})\} = P(\underline{x};\underline{A}_i) \qquad (2\text{-}12)$$

where the notation $P(\underline{x};\underline{A}_i)$ on the far right-hand side of (2-12) stresses the dependence of $P_i(\underline{x})$ upon the matrix of hyperplane coefficients, $\underline{A}_i$.

The generic form of our approximation, which is given by Equation (2-11), is seen to be implicitly dependent upon the weights $\{w_i; i = 1,\ldots, N_p+1\}$ and the hyperplane coefficients $\{A_i; i = 1,\ldots,N_p\}$. We shall refer to these aspects of the fit, for economy of notation, using the (N_p+1)-dimensional *weight vector*

$$\underline{W} = \left[w_1\ w_2 \cdots w_i \cdots w_{N_p+1}\right] \qquad (2\text{-}13)$$

and the overall *coefficient matrix*

$$\underline{A} = \begin{bmatrix} \underline{A}_1 \\ \underline{A}_2 \\ \vdots \\ \underline{A}_i \\ \vdots \\ \underline{A}_{N_p} \end{bmatrix} \quad ; (\underline{A}_i \text{ has dimension } Q_i \text{ x } (n+1)) \qquad (2\text{-}14)$$

Once the values of N_p and $\{Q_i = i = 1,2,...N_p\}$ are specified, the function $F(\underline{x})$ is specified fully by choosing $\underline{A}$ and $\underline{W}$. To emphasize this dependence we often write $F(\underline{x}; \underline{A}, \underline{W})$. The objective is now to determine $\underline{A}$ and $\underline{W}$ so that $F(\underline{x}; \underline{A}, \underline{W})$ provides a good approximation to the input/output data.

2.2.2 Mean-Squared Error Optimization Criterion

In order to finally determine $\underline{A}$ and $\underline{W}$ it is necessary to adopt some criterion of goodness for the resulting functional approximation. We shall use the optimization criterion of minimizing the mean-squared-error (MSE) of the approximation for the specified values of N_p (number of P-functions) and $\{Q_i; i = 1,2,...N_p\}$ (number of hyperplanes per P-function).

For M data samples the MSE may be expressed as

$$E = \frac{1}{M} \sum_{\ell=1}^{M} \left[y^{\ell} - F(\underline{x}^{\ell}; \underline{A}, \underline{W}) \right]^2 \qquad (2\text{-}15)$$

The minimization of E must be done numerically, and we shall see that the requisite computations can become quite costly for large values of N_p and Q_i. For this reason, we typically choose the number of P-functions and number of hyperplanes per P-function to be reasonably small during our first fitting attempts. Of course, if $N_p = 1$ we lose the ability to fit non-convex data; this can be a useful tool for discovering if the data is intrinsically convex. After examining the quality and characteristics of the fit, it is easier to make judgments about further increases in the complexity of the fit to be attempted. Perhaps surprisingly, even $N_p = 2$ or 3 with comparable values of Q_i often produces a totally acceptable fit because of the inherent adjustability of the approximating function.

Insofar as the minimization of E is concerned it turns out (see Section 2.2.3) that for any specified $\underline{A}$ there is a closed-form analytic solution* for the optimum set of weights, $\underline{W}$. The reason that an analytic solution for $\underline{W}$ is obtainable is that once $\underline{A}$ is specified, the P-functions are completely determined. $F(\underline{x})$ then depends only upon $\underline{W}$, and the dependence is linear. By way of contrast, $F(\underline{x})$ depends nonlinearly upon $\underline{A}$, and there is no closed-form analytic result for the $\underline{A}$ which minimizes E.

The fact that $\underline{W}$ is completely determined by $\underline{A}$ insofar as minimizing E is concerned, implies that the total number of "free parameters" that actually benefit the fit is equal to the total number of elements in the overall hyperplane coefficient matrix, $\underline{A}$, in Equation (2-14). Thus, the total number of free parameters in the overall fit is equal to

$$N_F = (n + 1) \sum_{i=1}^{N_P} Q_i \qquad (2\text{-}16)$$

For example, suppose the problem at hand requires fitting the relationship between 4 input variables and one output (dependent) variable. In this case $n = 4$. Even the simplest non-convex fit using two P-functions ($N_P = 2$) and two hyperplanes per P-function ($Q_1 = Q_2 = 2$) produces $N_F = 20$. This fit involves a numerical optimization over a 20-dimensional parameter space in order to minimize E. In subsequent sections we shall describe a practical computational procedure for accomplishing the requisite optimization. First, it is in order to set up the optimization equations.

---

*As opposed to requiring numerical minimization.

2.2.3 The Optimization Equations

The computational problem before us is to minimize the MSE, E, by suitably adjusting the collection of hyperplane coefficients, $\underline{A}$, and weights, $\underline{W}$.

Formally speaking, we may regard E as a function of $\underline{A}$ and $\underline{W}$ and denote this via the notation $E(\underline{A},\underline{W})$. It can be shown that E is differentiable with respect to the parameters $\underline{A}$ and $\underline{W}$. Thus, a necessary condition for a minimum of E is that the total derivative of E with respect to all of the parameters of $\underline{A}$ and $\underline{W}$ be equal to zero.

To elaborate, it is helpful to reassemble the elements of $\underline{A}$ and $\underline{W}$ into one overall row vector and to utilize the vector derivative or gradient concept. $\underline{A}$ has N_F elements and $\underline{W}$ has $(N_p + 1)$ elements. Thus, suppose we define the vector

$$\underset{\quad |\leftarrow N_F \rightarrow| \leftarrow N_p+1 \rightarrow|}{\underline{\gamma} = [\ \underline{\alpha}\ \vdots\ \underline{W}\ ] = [\gamma_1\ \gamma_2 \cdots \gamma_\mu \cdots \gamma_{(N_F+N_p+1)}]} \qquad (2\text{-}17)$$

where $\underline{\alpha}$ is an N_F-dimensional row vector composed of the elements of $\underline{A}$ and $\underline{W}$ is as defined in (2-13). In other words, there is a one-to-one correspondence between the elements of the row vector $\underline{\alpha}$ and the elements of the *matrix* of hyperplane coefficents, $\underline{A}$. The manner in which the elements of $\underline{A}$ are rearranged into the vector $\underline{\alpha}$ is arbitrary. Later we will offer one possible arrangement.

Now, strictly speaking E may be regarded as a function of the composite parameter vector $\underline{\gamma}$. Each specification of the elements of $\underline{\gamma}$ defines a point

in the overall parameter space, Γ, which has dimension $N_F + N_p + 1$. A minimum of E occurs only if the magnitude of the gradient of E (also called the "norm" of E) calculated in the Γ space is zero. We denote the gradient of E via the notation $\frac{\vec{dE}}{d\underline{\gamma}}$. This is a vector defined as follows:

$$\frac{\vec{\partial E}}{\partial \underline{\gamma}} = \underbrace{\frac{\partial E}{\partial \gamma_1}\vec{r}_1 + \frac{\partial E}{\partial \gamma_2}\vec{r}_2 + \cdots \frac{\partial E}{\partial \gamma_{N_F}}\vec{r}_{N_F} +}_{\frac{\vec{\partial E}}{\partial \underline{\alpha}}}$$

$$\underbrace{\frac{\partial E}{\partial \gamma_{N_F+1}}\vec{r}_{N_F+1} + \cdots + \frac{\partial E}{\partial \gamma_{(N_F+N_p+1)}}\vec{r}_{(N_F+N_p+1)}}_{\frac{\vec{\partial E}}{\partial \underline{W}}} \qquad (2\text{-}18)$$

where $\vec{r}_i$ denotes a unit vector in the $i^{\underline{th}}$ direction. As indicated in (2-18) the first N_F components of $\frac{\vec{\partial E}}{\partial \underline{\gamma}}$ comprise the gradient of E with respect to the components of $\underline{A}$, and the last $(N_p + 1)$ components of $\frac{\vec{\partial E}}{\partial \underline{\gamma}}$ comprise the gradient of E with respect to the elements of $\underline{W}$. In other words, $\frac{\vec{\partial E}}{\partial \underline{W}}$ is calculated holding $\underline{A}$ constant, and $\frac{\vec{\partial E}}{\partial \underline{\alpha}}$ is calculated holding $\underline{W}$ constant. Therefore, in order for $\left|\frac{\vec{\partial E}}{\partial \underline{\gamma}}\right|$ to be zero for arbitrary adjustments in $\underline{A}$ and $\underline{W}$ we require separately that

$$\left| \frac{\partial \vec{E}}{\partial \underline{\alpha}} \right| = 0 \qquad (2\text{-}19a)$$

and

$$\left| \frac{\partial \vec{E}}{\partial \underline{W}} \right| = 0 \qquad (2\text{-}19b)$$

Using Equations (2-11) and (2-15) it is possible to explicitly solve (2-19b) above for the optimum $\underline{W}$. In particular, let

$$g_{\ell i} = P_i(\underline{x}^{\ell}; \underline{A}_i) \quad ; \quad \begin{matrix} i = 1,\ldots,N_P \\ \ell = 1,\ldots,M \end{matrix} \qquad (2\text{-}20a)$$

$$\underline{G} = \overset{M}{\left\{ \begin{matrix} \\ \\ \\ \end{matrix} \right.} \overbrace{\begin{bmatrix} g_{\ell i} & \begin{matrix} 1 \\ 1 \\ \vdots \\ 1 \end{matrix} \end{bmatrix}}^{\leftarrow (N_P+1) \rightarrow} \qquad (2\text{-}20b)$$

That is, $\underline{G}$ is an M x $(N_p + 1)$ dimensional matrix composed of the P-functions evaluated at the various input data points, to which a column of 1's is adjoined. From (2-11) and (2-20a & 20b) we can define

$$F_\ell \equiv F_\ell(\underline{x}^\ell) \equiv \sum_{i=1}^{N_p} w_i \, P_i(\underline{x}^\ell) + w_{N_p+1} \quad \text{for } \ell = 1,\ldots,M \tag{2-21a}$$

$$\underline{F} = [F_1 \; F_2 \; \cdots \; F_\ell \; \cdots \; F_M] \tag{2-21b}$$

from which it follows that $\underline{F}$ can be expressed in the form

$$\underline{F} = \underline{W} \, \underline{G}^T \tag{2-22}$$

and E can be expressed in the form

$$E = \frac{1}{M} \left[\underline{y} - \underline{W} \, \underline{G}^T\right] \left[\underline{y} - \underline{W} \, \underline{G}^T\right]^T \tag{2-23}$$

Formally calculating $\frac{\vec{\partial E}}{\partial \underline{W}}$ yields:

$$\frac{\vec{\partial E}}{\partial \underline{W}} = \frac{2}{M} \left[\underline{W} \, \underline{G}^T \, \underline{G} - \underline{y} \, \underline{G}\right] = \frac{2}{M} \left[\underline{F} - \underline{y}\right] \underline{G} \tag{2-24}$$

Setting $\left|\frac{\vec{\partial E}}{\partial \underline{W}}\right| = 0$ results in the solution:

$$\underline{W} = \underline{y} \, \underline{G} \left[\underline{G}^T \, \underline{G}\right]^{-1} = \underline{W}(\underline{A}) \tag{2-25}$$

where we have emphasized the fact that the solution is dependent upon $\underline{A}$ (through $\underline{G}$) via the notation $\underline{W}(\underline{A})$. It can also be verified that $\frac{\partial^2 E}{\partial \underline{W}^2} > 0$; so the solution in (2-25) does, indeed, provide a minimum for E.

The solution for $\underline{W}$ given in (2-25) exists as long as the matrix $[\underline{G}^T\underline{G}]$ is invertible. One requirement for this to be true is that the number of sample points, M, used in the fit must be at least equal to $(N_p + 1)$. That is, (2-25) has a valid solution only if*

$$M \geq (N_p + 1) \qquad (2\text{-}26)$$

Since the number of P-functions comprising the fit is typically rather small, (2-26) is usually satisfied in practice.

If (2-26) is satisfied but $[\underline{G}^T\ \underline{G}]$ does not have an inverse, it is because two or more rows or columns of $[\underline{G}^T\ \underline{G}]$ are linearly dependent. If this occurs the problem can generally be overcome by reducing the number of P-functions and/or hyperplanes which are being tried in the fit.

It now remains to calculate $\frac{\partial \vec{E}}{\partial \underline{\alpha}}$. The magnitude of the μ^{th} component of this N_F-dimensional vector is obtained using (2-15):

$$\frac{\partial E}{\partial \alpha_\mu} = -\frac{2}{M}\sum_{\ell=1}^{M}[y^\ell - F(\underline{x}^\ell;\underline{A},\underline{W})]\frac{\partial F(\underline{x}^\ell)}{\partial \alpha_\mu} \quad ; \quad \mu = 1,2,\ldots,N_F \qquad (2\text{-}27)$$

*Actually a more important practical requirement is that $M \gg N_F$. This ensures that the data is not overfit by simply having an unjustifiably large number of parameters in the fitting function. Note from (2-16) that $M \gg N_F$ implies that (2-26) is satisfied.

In order to proceed further it is necessary to specify the correspondence between the elements of $\underline{\alpha}$ and the entries of the hyperplane coefficient matrix, $\underline{A}$.

Referring back to Equations (2-9) and (2-14) the following correspondence is suggested.

Take the elements of the matrix $\underline{A}_i$ (these are the hyperplane coefficients associated with the $i^{\underline{th}}$ P-function) and assemble them into a row vector, $\underline{\alpha}_i$. To do this let $\underline{\alpha}_i$ be assembled as shown below

$$\underline{\alpha}_i = \underbrace{\left[\underline{a}_1^{(i)} \right.}_{\substack{1^{\underline{st}} \text{ row of} \\ \underline{A}_i}} \underbrace{\left| \underline{a}_2^{(i)} \right.}_{\substack{2^{\underline{nd}} \text{ row of} \\ \underline{A}_i}} \cdots \underbrace{\left| \underline{a}_q^{(i)} \right.}_{\substack{q^{\underline{th}} \text{ row of} \\ \underline{A}_i}} \cdots \underbrace{\left| \underline{a}_{Q_i}^{(i)} \right]}_{\substack{\text{last row of} \\ \underline{A}_i}} \tag{2-28}$$

In other words $\underline{a}_q^{(i)}$ are the hyperplane coefficients of the $q^{\underline{th}}$ hyperplane of the $i^{\underline{th}}$ P-function assembled into an $(n + 1)$-dimensional row vector. Then $\underline{\alpha}_i$ collects all of the coefficients of the $i^{\underline{th}}$ P-function into a single row vector of dimension $[Q_i \cdot (n + 1)]$ by adjoining the partitions as indicated in (2-28) above.

Finally we form the overall $\underline{\alpha}$ vector by letting

$$\underline{\alpha} = \left[\underline{\alpha}_1 \mid \underline{\alpha}_2 \mid \cdots \mid \underline{\alpha}_i \mid \cdots \mid \underline{\alpha}_{N_p} \right] = \left[\alpha_1 \; \alpha_2 \cdots \alpha_\mu \cdots \alpha_{N_F} \right] \tag{2-29}$$

The vector $\underline{\alpha}$ is therefore partitioned into N_p subvectors in which the $i^{\underline{th}}$ partition comprises all of the coefficients of the $i^{\underline{th}}$ P-function. Equation (2-29) establishes the correspondence between the $\mu^{\underline{th}}$ component of $\underline{\alpha}$ and the appropriate P-function and hyperplane coefficient. It is now possible to evaluate (2-27).

Consider the $i^{\underline{th}}$ P-function. For any given input data point $\underline{x}^{\ell}$ and a trial solution for $\underline{A}_i$ we can evaluate (using the generalized form of Equation (2-8))

$$H_q^{(i)}(\underline{x}^{\ell}) = \underline{a}_q^{(i)} \cdot \underline{x}'^{T} \; ; \quad q = 1,2,\ldots Q_i \tag{2-30}$$

We can therefore identify the value, q_o, for which $H_q^{(i)}(\underline{x}^{\ell})$ is maximum over the range of q. From the <u>definition</u> of the $i^{\underline{th}}$ P-function (Equation (2-12)) and from the generic form of $F(\underline{x})$ given in (2-11) it follows that $\partial F(\underline{x}^{\ell})/\partial\alpha_{\mu}$ (which appears in (2-27)) is identically zero unless α_{μ} is one of the elements of the vector $\underline{a}_{q_o}^{(i)}$. Moreover, from (2-30) we obtain for the $k^{\underline{th}}$ element of $\underline{\alpha}_{q_o}^{(i)}$:

$$\frac{\partial F(\underline{x}^{\ell})}{\partial \alpha_{q_o k}^{(i)}} = w_i \, x_k^{\ell} \quad ; \quad k = 1,2,\ldots n \tag{2-31}$$

For other values of $q \neq q_o$, the derivative is zero.

Equation (2-31) holds for all values of i. Thus, to evaluate the quantities $\partial F(\underline{x}^{\ell})/\partial\alpha_{\mu}$ appearing in (2-27) one proceeds as follows:

(1) Choose a data point $\underline{x}^{\ell}$

(2) Calculate $H_q^{(i)}(\underline{x}^{\ell})$ for $i = 1,\ldots,N_p$

(3) For each i above, identify the value of q_o for which $H_{q_o}^{(i)}(\underline{x}^{\ell}) \geq H_q^{(i)}(\underline{x}^{\ell})$; $q = 1,2,\ldots,Q_i$
This identifies which row of $\underline{A}_i$ will contribute a nonzero result to the final derivative. More specifically it identifies which partition of $\underline{\alpha}_i$ in (2-28) will contribute a nonzero result.

(4) Utilize the association between α_μ and the partitions $\underline{\alpha}_i$ set forth in (2-28) and (2-29). This identifies the components $\partial F(\underline{x}^{\ell})/\partial\alpha_\mu$ which are identically zero. In addition, Equation (2-31) provides the values of the nonzero entries.

To obtain the desired gradient, we must evaluate (2-27) by repeating the above steps (1) - (4) for each of the M data points $\{\underline{x}^{\ell};\ \ell = 1,\ldots,M\}$. After having evaluated the components of $\frac{\partial \vec{E}}{\partial \underline{\alpha}}$ using (2-27) and using (2-24) for $\frac{\partial \vec{E}}{\partial \underline{W}}$, the norm of the gradient can be calculated in the conventional manner:

$$\left| \frac{\partial \vec{E}}{\partial \underline{\gamma}} \right| = \sqrt{\left| \frac{\partial \vec{E}}{\partial \underline{\alpha}} \right|^2 + \left| \frac{\partial \vec{E}}{\partial \underline{W}} \right|^2} = \sqrt{\sum_{\mu=1}^{N_F} \left(\frac{\partial E}{\partial \alpha_\mu}\right)^2 + \sum_{i=1}^{N_p+1} \left(\frac{\partial E}{\partial w_i}\right)^2} \qquad (2\text{-}32)$$

Observe in the above discussion that the resulting gradient is functionally dependent upon $\underline{W}$ and $\underline{A}$. However, we have already shown (Equation (2-25)) how to obtain the optimum $\underline{W}$ for a specified $\underline{A}$. Using (2-32) the norm of the gradient may be calculated for arbitrary trial values of $\underline{A}$ and $\underline{W}$. Moreover, the norm of the gradient can be expressed in terms of $\underline{A}$ alone if $\underline{W}$ is set at its optimum point. It can also now be appreciated that the gradient is not linearly related to $\underline{A}$; so that an analytic solution for the optimum $\underline{A}$ is not accessible.

2.2.4. Implementing the Optimization Procedure

As indicated in the previous sections we can numerically attempt to minimize the MSE of our functional approximation by searching for a zero in the norm of the gradient of the MSE with respect to the N_F free parameters in the hyperplane coefficient matrix, $\underline{A}$.

Any numerical solution, of course, is fundamentally unable to resolve the question of whether or not an absolute minimum (as opposed to a relative minimum) has been obtained. From a practical viewpoint, this is not a severe limitation because, in the final analysis the actual error magnitude will be the guide to acceptance of the approximation.

However, problems can occur if one attempts to blindly employ multidimensional gradient search techniques without appreciating the limitations imposed by computational resources.

Under constraints of finite computer resources it is important to start off the numerical optimization as close as possible to the optimum solution and to make judicious tradeoffs between suboptimum minimization efforts which consume fewer computer resources but can make significant progress towards a good solution.

In the next section of this report a computational procedure is detailed for implementing the optimization procedure.

Since the procedure is a computational one, and since it has been specifically implemented by software which itself is one of the deliverables under this effort, the implementation description will be unfolded in parallel with the software description.

III. DETAILED IMPLEMENTATION DESCRIPTION

3.1 OVERVIEW

The previous theoretical discussion will be specifically tied to a software implementation in this section of the report.

There are N_p P-functions, each with a given number of hyperplanes, and $N_p + 1$ weights to be determined so that the mean squared error (MSE) (defined in Section 2.2.2) of the piecewise-linear fit is minimized. Looked at from this point of view, then, there results a minimization problem. The method used to solve for the best hyperplanes and weights is to employ a gradient search. This choice is made because of the nonlinear dependence of the MSE upon the hyperplanes of the piecewise-linear fit. (See Section 2.2.3.)

Below will be described the normal sequence of operations in the software, assuming that the input is data from the Mobility Model consisting of M sample points, each one of which has a fixed number n, of input variables and a single output variable. Note that the actual data tape (or disc file, etc.) may have more than n input variables per sample point, but what is of interest is the actual number of input variables selected from each sample point, that are to be a part of the fit.

Throughout this section it is assumed that the following are available: the input/output data which is to be fit, the Structural Parameters determinining the number of P-functions, number of hyperplanes for each P-function, and the minimum acceptable activity level (see Section 3.1.5). Other Iteration Parameters (see Section 3.2.3) which determine the computational time and

effort which are to be devoted to the fit, are also supplied. Each of the P-functions is automatically initialized by the software, and then, using the gradient search scheme referred to above, is optimized to minimize the MSE within the limitations imposed on computer time and effort.

The optimization of the hyperplane coefficients and the weights occurs in three distinct parts: *Marginal Optimization, Sequential Optimization,* and *Joint Optimization.* These are described in Section 3.1. It is assumed in that section that all three will occur, as they do in normal program operation. Alternative modes of operation in which only some of the modes are used will be discussed in Section 3.2. Section 3.2 also discusses a mode in which no optimization is performed at all. Finally, in Section 3.3, are discussed the principal building blocks of all three types of optimization. These are P-function initialization, the optimizing of the weights (as a function of the hyperplane coefficients and the data), and the gradient search scheme.

3.1.1 Input and Standardization of the Data

The details of how to input the data are described in Section 4.2, but it should be emphasized that the lexicographic order of the input variables can be very important to the gradient search scheme. The reason for this will be evident in the description of the Initialization and Marginal Optimization of the P-functions (see Section 3.2.1). In any case, the first operation performed by the software is to put the data into a suitable "standard" form. This *standardization* consists of modifying the input variables so that each one of them has its mean subtracted off and is then divided by its standard deviation. This results in transformed input variables which are dimensionless. These dimensionless variables

are used to eliminate the problems of scaling and to express the input variables in units of standard deviations from the mean so that the different variables are commensurate.

Formally, the transformed input variables $\{\tilde{x}_k, k = 1,\ldots,n\}$ are related to the original input variables x_k by the formula,

$$\tilde{x}_k = \frac{x_k^{\ell} - m_k}{\sigma_k} \quad , \quad \ell = 1,\ldots,M \tag{3-1}$$

where

$$m_k = \frac{\sum_{\ell=1}^{M} x_k^{\ell}}{M} \quad \text{and} \quad \sigma_k^2 = \frac{\sum_{\ell=1}^{M} (x_k^{\ell} - m_k)^2}{M} \tag{3-2}$$

For ease of notation, the $\tilde{x}$'s will be referred to as x's in the following description of the program and its subroutine. That is, we shall always assume that the standardized data is being fitted and drop the "~" notation.

Because a numerical gradient search optimization is required, it is necessary to initialize the optimization routine before any optimization can be performed. In addition, for computational efficiency the optimization is carried out at a coarse level first, and proceeds to greater refinement in subsequent optimization stages. It is therefore natural to combine our discussion of initialization with this first optimization level, which is referred to as *Marginal Optimization.*

3.1.2 Initialization of the P-Functions and Marginal Optimization

Marginal Optimization is the first part of the three part optimization process mentioned earlier. In it, the following steps are taken: first a

single P-function P_1 is initialized (see Section 3.2.1) in subroutine INITA. Then weights w_1 and w' are chosen in the subroutine OPTW so that $w_1P_1(\underline{x}) + w'$ best fits the data (see Section 3.2.2). Then, in the subroutine OPT, a gradient search over the hyperplane coefficients (see Section 3.2.3) of this first P-function and weights w_1 and w' is performed in OPTW. In preparation for the next P-function, the evaluation of $w_1P_1(\underline{x}) + w'$ is subtracted from the output variable at each data point and entered in a residual array.

The steps for the second P-function are the same as those for the first, except that the quantity $w_2P_2(\underline{x}) + w''$ is fit to the residuals remaining at the end of the marginal optimization of $P_1(\underline{x})$. Initialization, initial computation of the weights w_2 and w'', optimization over the coefficients of the hyperplanes of $P_2(x)$, w_2, and w'' and a final recalculation of w_1, w_2, and w'' are performed. Then the residuals

$$y^{\ell} - w_1P_1(\underline{x}^{\ell}) - w_2P_2(\underline{x}^{\ell}) - w'', \ \ell = 1,\ldots,M$$

are calculated in preparation for the third P-function.

This process continues until all N_p P-functions have been involved one at a time. At each step of this marginal fitting procedure, the residual that has not been fit at the preceding step is fit during the current step to the best extent possible using a single P-function present alone. More details pertinent to Marginal Optimization are found in Section 3.2.1. The resulting P-function coefficients and weights are then used as the starting point of the next finer level of optimization: *Sequential Optimization.*

3.1.3 Sequential Optimization

In Sequential Optimization, the strategy is similar to Marginal Optimization in that we fit the residual functional variation not yet accounted for, and the fit is obtained by sequentially optimizing one P-function at a time. However, in this stage of optimization all of the desired P-functions are present while the fit is adjusted one P-function at a time. (This is to be contrasted with Marginal Optimization in which, at any step, the function being fit to the residual consists of a single P-function.) The procedure is as follows.

We start with the P-function coefficients and weights obtained from Marginal Optimization. Then we may assume that the P-functions $P_1,\ldots,P_N$ and weights $w_1 \cdots w_{(N_p+1)}$ are available. The residual formed by eliminating P_1:

$$y^{\ell} - (w_2 P_2(\underline{x}^{\ell}) + \cdots + w_{N_p} P_{N_p}(\underline{x}^{\ell}) + w_{N_p+1}) \qquad (3\text{-}3)$$

is calculated for each data point $(\underline{x}^{\ell},y^{\ell})$ for $\ell = 1,\ldots,M$.

Then $P_1(\underline{x})$, w_1 and w' are optimized in the subroutine OPT so that

$$w_1 P_1(\underline{x}) + w'$$

best fits the residual. This defines a new w_1 and replaces the old value of w_{N_p+1} with the constant weight $w_{N_p+1} + w'$.

The next step is to evaluate the residuals formed by eliminating P_2:

$$y^{\ell} - (w_1 P_1(\underline{x}^{\ell}) + w_3 P_3(\underline{x}^{\ell}) + \cdots + w_{N_p} P_{N_p}(\underline{x}^{\ell}) + w_{N_p+1}) \qquad (3\text{-}4)$$

for $\ell = 1,\ldots,M$ and optimize the choice of the coefficients of $P_2(\underline{x})$, w_2 and w_{N_p+1}.

This procedure continues sequentially through the P-functions, as long as none of the five stopping criteria is met (computer time, RMS error, error change, step size, gradient magnitude - see Section 3.2.3). If no stopping criteria has been met at the last P-function, the Sequential Optimization will start again with the first P-function and continue sequentially through all the P-functions. The computer time limit will force the process to stop if one of the other four criteria doesn't stop it first. This brings us to *Joint Optimization.*

3.1.4 Joint Optimization

This third step of optimization proceeds from the results of the prior optimization steps. We may therefore assume that P-functions and weights have been previously defined. The procedure is simpler to describe, since all the P-functions and all the weights are simulataneously changed in the gradient search. The gradient search over all the hyperplane coefficents and weights occurs in the subroutine OPT. Following that, the subroutine OPTW is called to exactly determine the optimum weights for the coefficients.*

Joint optimization is a more ambitious and computationally costly procedure unless one is reasonably close to the solution for the optimum hyperplane coefficients. This is because the number of possible directions that may have to be tried increases factorially with the number of degrees of

*The gradient search is computationally more efficient than repetitive numerical evaluation of the exact analytical solution for $\underline{W}$. The analytic evaluation of $\underline{W}$ is therefore always reserved for the final evaluation of $\underline{W}$ after no further changes in $\underline{A}$ occur.

freedom. For this reason, the Marginal and Sequential optimization, in which the number of degrees of freedom is much smaller, are used in an attempt to get a good starting point for the Joint optimization.

Each of these optimization stages has adjustable stopping critera in terms of computational time, RMS error, error change, gradient step size and norm of gradient, which are used to control the amount of effort spent in the various optimization stages. (With the exception of the time limits, which *must* be separately specified for each type of optimization, the remaining stopping criteria may either be separately specified; or by default the same criteria is used for all stages of optimization.)

3.1.5 Removing Inactive Hyperplanes

In the course of the optimization, the activity level of each hyperplane is computed. The activity level of the $q^{\underline{th}}$ hyperplane is the number of sample points on which that hyperplane is actually the one on which the maximum over q of $\underline{a}_q^{(i)} \cdot \underline{x}'$ occurs for the $i^{\underline{th}}$ P-function. An input supplied to the program is the minimum acceptable activity level per hyperplane. If a hyperplane falls below that level it is eliminated. This is a safeguard against an overly ambitious and wasteful specification of the number of hyperplanes.

3.1.6 Returning to Original Coordinates

At this point a fit has been made between the output variable and the standardized input variables. Returning to the "~" notation of Section 3.1.1, the N_P P-functions at the end of optimization may now be called $\tilde{P}_1(\underline{\tilde{x}}),\ldots,$ $\tilde{P}_{N_P}(\underline{\tilde{x}})$ and the weights may be denoted $\tilde{w}_1\ldots,\tilde{w}_{N_P+1}$ to represent the fact that the weights and P-functions now chosen, operate on the standardized input variables. The approximation achieved is:

$$F(\underline{\tilde{x}}) = \tilde{w}_1\tilde{P}_1(\underline{\tilde{x}}) + \cdots + \tilde{w}_{N_P}\,\tilde{P}_{N_P}(\underline{\tilde{x}}) + \tilde{w}_{N_P+1} \qquad (3\text{-}5)$$

The program defines new P-functions, $P_i(x)$, which operate on the unstandardized (i.e., original input) data so that

$$P_i(\underline{x}) = \tilde{P}_i(\tilde{\underline{x}}),\ i = 1,\ldots,N_p \tag{3-6}$$

That is the new P-functions, P_i, operating in the un-standardized input variables give the identical evaluation as the P-functions coming from the optimization procedures, $\tilde{P}_i$, operating on the standardized variables. It is evident that if we leave the w_i as they are and substitute (3-6) into (3-5) there results

$$F(\tilde{\underline{x}}) = \sum_{i=1}^{N_p} \tilde{w}_i\ \tilde{P}_i(\tilde{\underline{x}}) + \tilde{w}_{N_p+1} = \sum_{i=1}^{N_p} \tilde{w}_i\ P_i(\underline{x}) + \tilde{w}_{N_p+1} = F(\underline{x})$$

Equation (3-6) is attained by determining the appropriate new P-functions (i.e., hyperplane coefficients) which operate on the unstandardized input variables. Let $\underline{a}_q^{(i)}$ represent the unstandardized $(n + 1)$-dimensional vector of hyperplane coefficients for the $q^{\underline{th}}$ hyperplane of the $i^{\underline{th}}$ P-function, and let $\tilde{\underline{a}}_q^{(i)}$ denote the corresponding standardized coefficients which resulted from the optimization procedures. The objective in defining unstandardized coefficients $\underline{a}_q^{(i)}$ is to have

$$\sum_{k=1}^{n} a_{qk}^{(i)}\ x_k + a_{q,(n+1)}^{(i)} = \sum_{k=1}^{n} \tilde{a}_{qk}^{(i)}\ \tilde{x}_k + \tilde{a}_{q,(n+1)}^{(i)} \tag{3-7}$$

i.e., to have the dot product $\underline{a}_q^{(i)} \cdot \underline{x}'^{T}$ equal to the dot product $\underline{\tilde{a}}_q^{(i)} \cdot \underline{\tilde{x}}'^{T}$. Consider the following relationship between standardized and unstandardized coefficients. For the $q^{\underline{th}}$ hyperplane of the $i^{\underline{th}}$ P-function the unstandardized coefficients are $\{a_{qk}^{(i)};\ k = 1,2,\ldots(n+1)\}$. Similar notation with the "~" superscript denotes the standardized coefficients. Let

$$\left.\begin{aligned} a_{qk}^{(i)} &= \frac{\tilde{a}_{qk}^{(i)}}{\sigma_k} \\ a_{q,(n+1)}^{(i)} &= \tilde{a}_{q,(n+1)}^{(i)} - \sum_{k=1}^{n} \frac{\tilde{a}_{qk}^{(i)} m_k}{\sigma_k} \end{aligned}\right\} \qquad (3\text{-}8)$$

where σ_k and m_k are defined in Section 3.1.1. It can readily be verified that if the Equations of (3-8) are substituted in the left-hand side of (3-7), the right hand side of (3-7) results. Using (3-8) the functional approximation can be expressed in terms of the unstandardized hyperplane coefficients, and they would apply to unstandardized independent variables.

3.2 ESSENTIAL INGREDIENTS OF THE THREE TYPES OF OPTIMIZATION

In the previous section, the three types of optimization: Marginal, Sequential, and Joint were described. In Marginal Optimization, an initialization of the P-functions is required. This ingredient of Marginal Optimization is described in Section 3.2.1.

In all three types of optimization, the weights are optimized before and after the gradient search is performed. This ingredient of optimization is described in Section 3.2.2.

Finally, the gradient search itself is described in Section 3.2.3.

3.2.1 P-Function Initialization (Used in Marginal Optimization)

In Marginal Optimization we attempt to fit the input/output data using one P-function at a time (see Section 3.1.2). Ideally, we would begin (initialize) this fit by specifying a matrix of hyperplane coefficients $\underline{A}_i$ for each of the P-functions $\{P_i(\underline{x}); \underline{A}_i; i = 1,2,\ldots,N_p\}$. To specify the $\underline{A}_i$ we should ideally account for the interdependencies between the n input variables (the components of $\underline{x}$).

To avoid the complexities of examining such interdependencies we choose an initialization procedure which can be performed *marginally*; i.e., one-dimension at a time. Thus, as we encounter each of the P-functions during Marginal Optimization, it will be initialized over only one-dimension. This is equivalent to setting all but two of the columns of $\underline{A}_i$ equal to zero [see Equation (2-9)]. One of the two nonzero columns comprises the hyperplane coefficients* associated with the input variable being initialized over. The other nonzero column contains the required constant terms to complete the P-function specification. This is described more fully below.

The first P-function is initialized upon the first input variable, the second P-function upon the second, etc. If there are more P-functions than input variables, and if there are n input variables, the (n + 1) P-function is also initialized on the first input variable. In general, the $i^{\underline{th}}$ P-function is initialized on the $k^{\underline{th}}$ input variable, where i is equivalent to k mod n and k is between 1 and n.

It is important to note that this method of initialization fits the input variables in the order they are furnished. Moreover, if there are

*In this case the hyperplanes are line segments and the weights are the slopes of the line segments.

fewer P-functions than input variables ($N_p < n$), then only the first n input variables that are encountered will be involved in the Marginal Optimization stage. It has been found that better fits are usually achieved (particularly when $N_p < n$) when the independent variables are arranged in a particular order. Specifically, it has been found that it is useful to arrange the input variables in decreasing order of the absolute value of the correlation between those variables and the output variable. (See Section 4.1.1 for more explicit details.)

Since any single P-function presented alone is convex, and since our Marginal Optimization stage attempts the fit using only one P-function at a time, it is reasonable to initialize the P-functions so that they start out being convex. The choice is somewhat arbitrary, and we shall base our initialization upon a semicircular function. That is, our initialization will be chosen so that a semicircle is approximated by a continuous piecewise-linear function in which the number of line segments (recall that Marginal Optimization initializes in one-dimensional space where hyperplanes are simply line segments) of the fit is equal to the number of hyperplanes selected for the given P-function. The semicircle that is chosen has a radius of 2 and is situated as shown in Figure 5. (The radius of the semicircle can be kept fixed regardless of the actual range of the input variable because the optimization is performed using the standardized form of the input data (Section 3.1.1.) The value of 2 was chosen for the radius because most of the input variable points should lie within 2 standard deviations of the mean.)

The initial continuous piecewise-linear approximation to the semicircle can be described as follows. Suppose initialization is to be performed

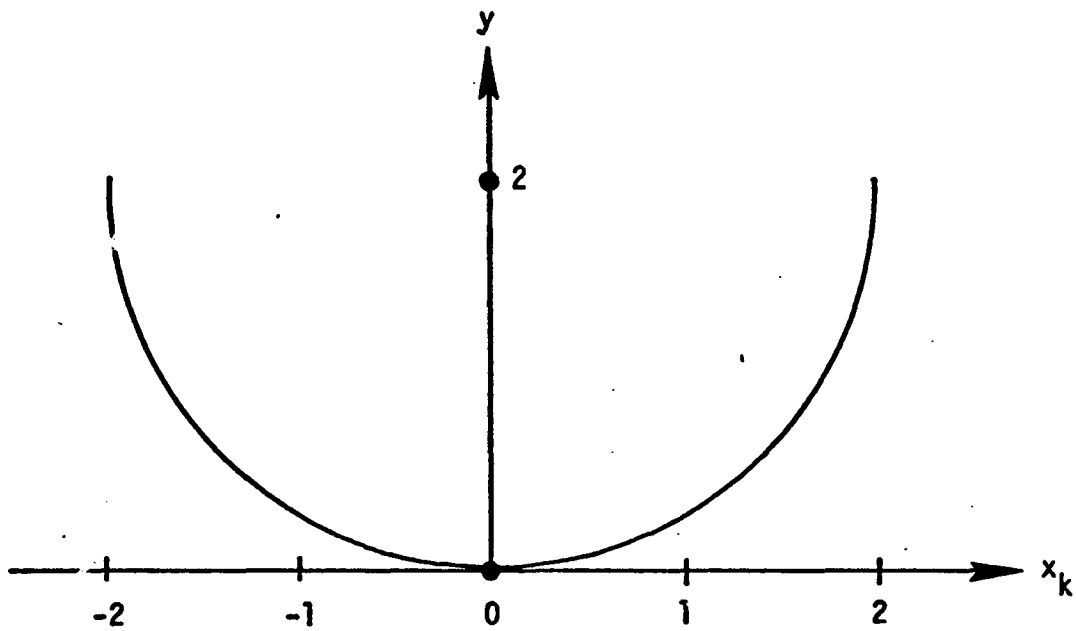


Figure 5. Basic Form of Initializing Function (Optimization Initialized Over the Variable x_k)

over the input variable x_k. The interval $-2 \leq x_k \leq +2$ is divided into NHYP(i) = Q_i parts, each of length $4/Q_i$. Projecting these points, which lie along the x_k axis, up to the semicircle defines Q_i points on the semicircle. Connecting these points with straight line segments provides the desired initial fit to the semicircle. The resulting fit for $Q_i = 4$ is shown in Figure 6. Mathematically the procedure can be described as follows:

Let

$$x_0 = -2$$
$$x_1 = -2 + 4/Q_i$$
$$\vdots$$
$$x_q = -2 + i(4/Q_i)$$
$$\vdots$$
$$x_{Q_i} = 2$$

and let

$$y_q = -1 + \sqrt{1 - x_q^2}$$

Then define the $q^{\underline{th}}$ hyperplane's coefficients as follows:

$$a_{qk}^{(i)} = 0 \text{ if } k \neq i \text{ or } k \neq n + 1$$

$$a_{qk}^{(i)} = \frac{y_{q+1} - y_q}{x_{q+1} - x_q}$$

$$a_{q,n+1}^{(i)} = \frac{x_{q+1}\, y_q - y_{q+1}\, x_q}{x_{q+1} - x_q}$$

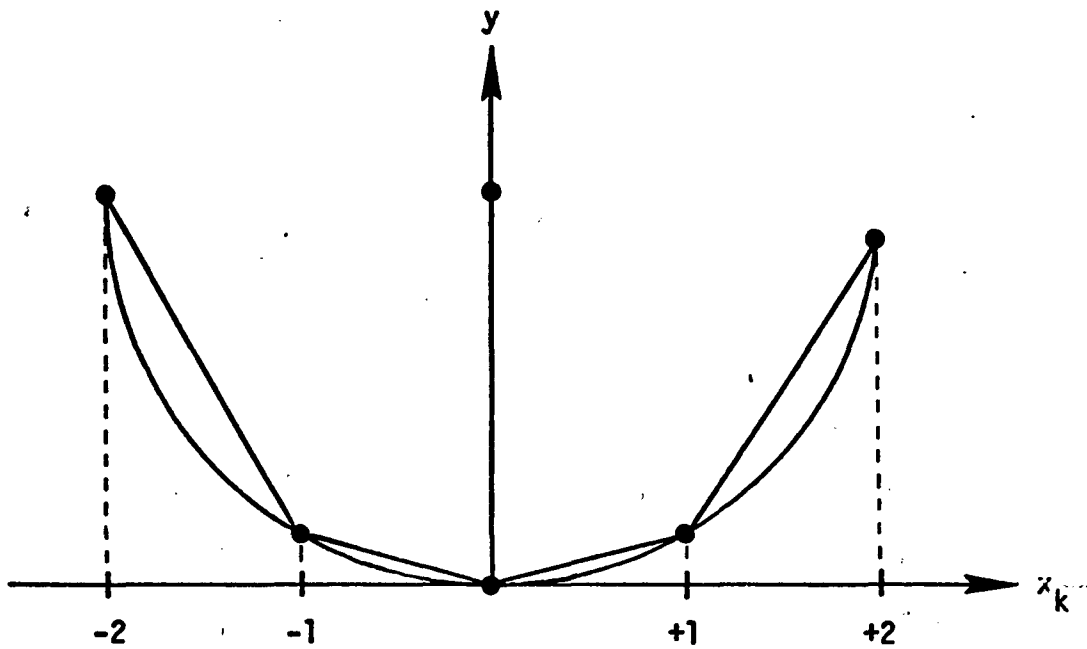


Figure 6. Initial COMPLIAR Fit For Q = 4

Thus, for the $q^{\underline{th}}$ hyperplane, the $k^{\underline{th}}$ coefficient is the slope between the $q^{\underline{th}}$ and q + 1 points on the semicircle, and the (n + 1) component is the y intercept of the line connecting these same two points on the semicircle.

3.2.2 Optimization of the Weights

Suppose that at a given point the coefficients for P-functions 1,..., N_p have been determined. The objective is to determine the weights w_1, w_2,..., w_{N_p+1} in (2-13) so that the MSE (Equation (2-15)) is minimized.

The solution for the optimum weights is given by Equations (2-20) - (2-25).

3.2.3 Gradient Search Scheme

The gradient search technique is used in all three forms of optimization. Following is a description of the scheme for Joint Optimization. For Marginal and Sequential Optimization, the vector, $\underline{\gamma}$ (see (2-17)) consists only of the one P-function's coefficients which are being optimized along with two weights: one is the weight of the P-function being optimized, and the other is a constant weight (see Section 3.1.2 and 3.1.3). First all the coefficients of all the hyperplanes of the P-function(s) being optimized, along with the weights are arranged in one long array, $\underline{\gamma}$, as in Equation (2-17). Since the function being optimized is E, the gradient of E with respect to $\underline{\gamma}$ is computed. This gradient is denoted $\frac{\vec{\partial E}}{\partial \underline{\gamma}}$. Denote the initial set of hyperplane coefficients and weights $\underline{\gamma}_o$. Let $\Delta\underline{\gamma}$ denote the change in the value of $|\underline{\gamma}_o|$; i.e., the step size. Then the trial value of $\underline{\gamma}$ is*

$$\underline{\gamma}_{trial} = \underline{\gamma}_o - \frac{\Delta\gamma}{m} \frac{\vec{\partial E}}{\partial \underline{\gamma}}$$

*Remember that $\underline{\gamma}$ is a vector in a space of dimension $N_F + N_p + 1$.

where $m = |\partial E/\partial \underline{\gamma}|$. We also define a step size adjustment factor, k_γ (supplied as an input by the user). If $\underline{\gamma}_{trial}$ actually achieves a smaller error, E, then it is taken as the new state variable; replacing $\underline{\gamma}_0$, and the procedure is repeated. If the trial vector achieves a larger error, E, then the step size, $\Delta\gamma$, is <u>divided</u> by k_γ and a new trial vector is calculated. On the other hand, if a certain number (NSMX) of decreases of E occur in a row (NSMX is supplied as an input by the user), then the step size is <u>multiplied</u> by k_γ. (The details of how the step size is increased and decreased can be found in the IFTRAN commented listing of the subroutine OPT. See Section 6.3.3., and in particular, Figure 11.)

In any case, the process continues to iterate with five stopping criteria (listed in the order they are printed in the output) checked at each iteration:

(1) Computer time

(2) RMS error

(3) Error change (the difference between the current RMS error and the previous RMS error)

(4) Step size

(5) Gradient magnitude.

The stopping criterion of time must be supplied by the user; the four others may be supplied and are otherwise given nominal values by the program.

To complete the definition of the gradient search, the calculation of the gradient $\frac{\partial \vec{E}}{\partial \underline{\gamma}}$ must be defined. The following implements the procedure discussed in Equations (2-27) - (2-32).

The first N_F elements of the array $\underline{\gamma}$, are hyperplane coefficients, $a_{qk}^{(i)}$. The partial derivative of E with respect to any $a_{qk}^{(i)}$ may be written (using (2-27) and (2-31)) as

$$\frac{\partial E}{\partial a^{(i)}_{qk}} = \frac{2}{M} \sum_{\ell=1}^{M} (F(\underline{x}^{\ell}) - y^{\ell}) \cdot \begin{cases} w_i x_k & \text{if } q^{\underline{th}} \text{ hyperplane is evaluated on } \ell^{\underline{th}} \text{ point} \\ 0 & \text{otherwise} \end{cases}$$

The partial derivative with respect to a weight, w_i, $i = 1,\ldots,N_p$ is (using (2-20) through (2-23))

$$\frac{\partial E}{\partial w_i} = \frac{2}{M} \sum_{\ell=1}^{M} (F(\underline{x}^{\ell}) - \underline{y}^{\ell})) \; P_i(\underline{x}^{\ell}) \quad ; \quad i = 1,\ldots,N_p$$

$$\frac{\partial E}{\partial w_{N_p+i}} = \frac{2}{M} \sum_{\ell=1}^{M} (F(\underline{x}^{\ell}) - y^{\ell})$$

3.3 ALTERNATE MODES OF PROGRAM OPERATION

The normal sequence of program operation, where the P-functions are initialized and optimized by the program, has been described in Sections 3.1 and 3.2. Other options of program operation are available and are described below. They include:

- calculate statistics on the input data with no fitting performed (3.3.1)
- refine previously determined P-functions (3.3.2)
- calculate net hyperplane statistics from previously determined P-functions (3.3.3).

It is important to note that the details of instrumenting all of the alternate modes described in this section are described in Section 4.

3.3.1 Calculate Statistics on Input Data

There exists a program option to calculate the mean, standard deviation, maximum, and minimum for each of the input variables, as well as the output variable. Using this same option, the user also obtains the correlation matrix for the collection of the n input variables and the output variable. This option includes a sub-option which provides a listing of all the data points.

This option was included in the program to allow the user to do some initial checks on the data and to help the user in the choice of lexicographic order of input variables (see Section 3.2.1 on Initialization). The normal rule of thumb is to order the variables in decreasing magnitude of their correlation with the output variable. One might wish to deviate from this guideline, however, if, e.g., two input variables are highly correlated with each other.

3.3.2 Refine Previously Determined P-Functions

Suppose a given set of P-functions is found to give a reasonably acceptable model but it is desired to use the program to further optimize these P-functions to obtain a better model. In other words, suppose there exists a set of P-functions which is to be used as a starting point in the optimization scheme. This type of scenario may be broken into two distinct cases.

In the first case, the number of P-functions and the number of hyperplanes in each P-function is kept the same, and one simply initializes a new run with these prior results. This is a useful capability when one desires to examine the benefits of adding more data points and/or allowing more time

in the various stages of optimization. In this first case, the program writes out the P-functions (specifically, their hyperplane coefficients, the weights, the number of input variables, and the number of hyperplanes for each P-function) in binary format at the end of execution and begins execution on a subsequent run using these parameters. The hyperplane coefficients that are available in this first case are in non-standardized form.

In the second case, the hyperplane coefficients $a_{qk}^{(i)}$ of the P-functions, the weights, the number of P-functions, N_p, and the number of hyperplanes in each P-function, Q_i, $i=1,\ldots,N_p$, are input by the user through the STPARM parameter list. The user must furnish the hyperplane coefficients in non-standardized form. Thus, when running from previously defined P-functions (in which case Initialization is bypassed), the program assumes that the P-functions are input in non-standardized form. Before the program proceeds to optimize these P-functions, it puts them in standardized form. This is accomplished using Equation (3-8). The resulting P-functions are therefore standardized in the same fashion as if automatic initialization had been employed.

An important example of the second case mentioned above occurs when an attempt is made to improve a previously obtained fit, by trying an additional P-function. In this situation one has available the non-standardized P-function coefficients of the prior fit from the printed output of the run.* These P-function coefficients are utilized as the initial coefficients for the new fit; but the initial coefficients for the additional P-function must also be supplied.

*These are the results contained in Table 1B of the output described in Section 4.4.

The added P-function can be initialized any number of ways by the user. In the absence of any compelling reason to do otherwise, the user may choose to manually initialize this P-function using the same strategy as employed in Automatic Initialization (see Section 3.2.1). Thus, suppose we wish to initialize the new P-function (call it P_i) on the $k^{\underline{th}}$ variable, x_k. Suppose, also, that there are to be Q_i hyperplanes in this P-function. Then, as in Section 3.2.1, the initial values of the coefficients of the Q_i hyperplanes comprising P_i can be defined as follows ($q=1,\ldots,Q_i$)

$$\left.\begin{aligned} \tilde{a}^{(i)}_{qk} &= 0 \text{ if } k \neq i \text{ or } k \neq n+1 \\ \tilde{a}^{(i)}_{qk} &= \frac{y_{q+1} - y_q}{y_{q+1} - x_q} \\ \tilde{a}^{(i)}_{q,n+1} &= \frac{x_{q+1}\, y_q - y_{q+1}\, x_q}{x_{q+1} - x_q} \end{aligned}\right\} \qquad (3\text{-}9)$$

Note that these are the desired initial coefficients , $\tilde{a}^{(i)}_{qk}$, of P_i to operate on <u>standardized</u> variables. The program, however, assumes that the input hyperplane coefficients are <u>not</u> standardized; so we must supply the coefficients of the hyperplanes comprising P_i so that, after standardization, they satisfy Equation (3-9).

Consider setting the input non-standardized coefficients $a_{qk}^{(i)}$ to

$$\left.\begin{aligned} a_{qk}^{(i)} &= 0 \text{ if } k = i \text{ or } k \neq n+1 \\ a_{qk}^{(i)} &= \tilde{a}_{qk}^{(i)} / \sigma_k \\ a_{q,(n+1)}^{(i)} &= \tilde{a}_{q,(n+1)}^{(i)} - \frac{m_k}{\sigma_k} \tilde{a}_{qk}^{(i)} \end{aligned}\right\} \quad (3\text{-}10)$$

Where the values $\tilde{a}_{qk}^{(i)}$ are obtained from Equation (3-9) and the m_k and σ_k are as described in Section 3.1.6. Then it can readily be verified that after the $a_{qk}^{(i)}$ have been standardized, the result will be the desired coefficents $\tilde{a}_{qk}^{(i)}$.

IV. USE OF THE PROGRAM

This section contains detailed instructions on the use of the COMPLIAR software, including a description of inputs required and examples of outputs furnished by the program.

Section 4.1 will describe how to use the program in a step-by-step fashion. Section 4.2 contains a complete list of program inputs together with their definitions. Before using the program, it is recommended that both Sections 4.1 and 4.2 be read to get a complete idea of how to use the program.

Section 4.3 describes common problems which may be encountered in program operation and also contains a list of error messages produced by the program. The printed output of the program is described in Section 4.4. Finally, the output of P-functions is described in Section 4.5.

4.1 STEP-BY-STEP PROGRAM USE

Below is a description of typical ways of using the COMPLIAR software to analyze data and to produce the desired multidimensional fit.

4.1.1 Initial Data Analysis Alone

By setting NPFUNC = 0, the program will take the input data (see Tables 4.1 and 4.2) and compute and print summary properties of the data. The minimum, maximum, mean, and standard deviation are calculated for each of the designated input variables and the output variable. Also, the correlation matrix is calculated for the (n + 1)-dimensional vector composed of the n input variables and the output variable. If NPFUNC = -1, in addition to the above, a listing of all of the data points is provided.

One of the purposes of this Data Analysis option is to familiarize the user with the salient characteristics of the data. For example, a listing of the data points permits identification of outliers. The correlation matrix provides an indication of the degree of coupling which exists between the various input variables and also between each of the input variables and the output variable. The information contained in the correlation matrix is important to the working of the program and can influence how good the ultimate fit will be (see Sections 3.1.2 and 3.1.3). For example, for a given amount of effort spent in the various stages of optimization, it is advisable to arrange the input variables in decreasing order of importance. This is usually accomplished by arranging the input variables in decreasing order of the absolute value of their correlations with the output variable. This often works well, but must be tempered by one's knowledge of the problem at hand. It must be remembered that the correlation is a measure of the linear association between two variables. If the correlation is near unity in absolute value, then the two variables are closely (and linearly) related. However, the correlation can be small and the variables can still be closely (but not linearly) related. Examples can easily be constructed where a quadratic functional relationship exists between two variables, yet the correlation between them is 0. Therefore, if there is even a strong belief that a variable is important, it is useful to place it near the top of the list.

Also, an examination of the correlation matrix may reveal a high correlation between two input variables, both of which have high correlation with the output variable. In this case, most of the information should be

obtainable from one of the two variables. Hence, a good strategy is to include one of them near the top of the list; but to put the other one at the bottom.

4.1.2 Initial Run of the Program

Having decided on an order for the input variables, the next step is to run the program and examine the structure and quality of the fit obtained. Unless there is a-priori knowledge that the realtionship between input and output is convex, at least two P-functions should be used. A good rule of thumb is to try two P-functions with two hyperplanes per P-function and another run with three hyperplanes per P-function. After some experimentation, it becomes clearer how many P-functions and hyperplanes per P-function are necessary for a good fit. The activity vector may be used as an indicator of the benefit of additional hyperplanes. If adding additional hyperplanes still leaves a reasonably high level of activity on most hyperplanes, then the number of hyperplanes per P-function is probably appropriate. However, if many of the hyperplanes show up with a very low activity, or even 0 activity, then the complexity of the P-functions, or their number, or both, is probably too large.* Experimentation is required in order to determine reasonable time limits for the three types of optimization, as well as values for the other optimization termination criteria.

4.1.3 Starting Off the Optimization Process from Previously Determined P-Functions

As outlined in Section 3.3.2, there are basically two ways of starting off the program with previously determined P-functions.

---

*At least for the amount of data supplied.

The first is basically a start/restart mechanism in the program: the P-functions, and weights along with other essential parameters can be saved at the termination of a run of the program by specifying an appropriate save unit, ISAVE. On another run, the program can read the old P-function from unit IOLD, where IOLD is identified as the aforementioned unit. When the P-functions and weights are saved and restored in this fashion, they are in binary format. One may then restart the program and proceed directly to either Sequential Optimization or Joint Optimization. The main utility of this mode is to enable one to stop the program midway through a convergence to a P-function and to have the program compute the hyperplane statistics. It is important to note that when restarting the program from P-functions and weights saved in this fashion, the same P-function structure, i.e., number of P-functions and number of hyperplanes in each P-function, must be employed as were used in the initial run. In fact, the following parameters will be saved and restored when using this option:

- The number of input variables
- The total number of hyperplanes
- The number of P-functions
- The number of hyperplanes in each P-function
- The hyperplane coefficients
- The P-function weights.

The second method of starting from previously determined P-functions involves inputting the P-functions directly (in decimal format) via the STPARM parameter list. This option allows the user to start from a prior

fit and proceed with optimization while simultaneously changing the P-function structure. The details of how to initialize the new P-functions are given in Section 3.3.2.

4.2 PROGRAM INPUT PARAMETERS

Below is a table of the input parameters for the COMPLIAR program. They are in the form of parameter lists. These lists are described in the same order they must be supplied to the program and are described in Tables 4.1 through 4.5. The format in each table is the same: the first column contains the name of the variable; the second column contains a default value, if one exists; or a "*" if a value must be furnished by the user; the third column contains a definition and/or explanation of the variable.

Table 4.1 DD PARM Namelist (Data Definition)

Variable Name	Default Value	Definition
NVAR	*	Actual number of variables in each record of the data file, including both input and output variables. (Note: the input variables involved in the fit may be a subset of the total set supplied. See NOFS.)
NOFS	*	Number of input variables involved in the fit. (This is the parameter n referred to in Equation (2-1) of the text.)
LOC	*	An array specifying the locations, within the NVAR input and output variables, of the NOFS selected input variables. (Example: if NVAR = 9, NOFS = 4, LOC = 5,6,7,8, then there will be four "input" variables and their locations on each record will be in the 5th, 6th, 7th, and 8th positions, respectively of NVAR.)
LOCDEP	*	Location, amongst the NVAR variables, of the "output" (dependent) variable.
JVAR	*	An array specifying the names to be given to the NOFS variables specified in the same order as in LOC. The names must be specified as Hollerith data and must be no longer than 8 characters.
JVARD	*	The name of the dependent variable, which must be specified as Hollerith data and be no longer than 8 characters.

*This symbol indicates that no default is available. The variable must be supplied as an input by the user.

Table 4.2 INPARM Namelist (Input Parameters)

Variable Name	Default Value	Definition
NSAMP	*	Maximum number of data points supplied to program. (This has been denoted by M in the accompanying text. See Equation 2-4.)
IVERY	1	When data is read, every IVERY point is used (Example: when IVERY = 3, every third point is read).
IUNIT	1	Unit from which data is read.
IOLD	0	When non-zero, P-functions in binary format are read from this unit. The following variables are read: number of "input" variables, the total number of hyperplanes, the number of P-functions, the number of hyperplanes in each P-function, the hyperplane coefficients and weights. When P-functions are to be input from STPARM Namelist, IOLD must remain 0.

*This symbol indicates that no default is available. The variable must be supplied as an input by the user.

Table 4.3 STPARM Namelist (Structure Parameters)

Variable Name	Default Value	Definition
NPFUNC	*	When NPFUNC > 0, NPFUNC is the number of P-functions. This has been denoted by N_P in the accompanying text. See Equation (2-11). When NPFUNC = 0, the program processes the data, producing summary characteristics for the input/output data including: the minima, maxima, means, and standard deviations; also produced is a matrix of correlations. When NPFUNC = -1, the above is produced, along with a listing of all the input/output data supplied.
NHYPS	*	If non-zero, this is the number of hyperplanes per P-function, and all P-functions have the same number of hyperplanes. If zero, then the number of hyperplanes in each P-function may be different and is input via the array NHYP.
NHYP	(required only if NHYPS = 0)	This is an array giving the number of hyperplanes in each P-function Example: If NPFUNC = 3, NHYPS = 0, and NHYP = 1, 3, 7, then there will be one hyperplane in the first P-function; 3 hyperplanes in the second P-function; and 7 hyperplanes in the third P-function.
MINACT	1	This is the minimum allowed activity for each hyperplane. If a hyperplane has less than this level of activity, it will be eliminated.
HYP	(required only if hyperplanes are to be read from namelist)	This is an array consisting of the hyperplane coefficients in decimal format. This requires IOLD to be 0. The hyperplane coefficients input must be consistent with NOFS, NPFUNC, NHYPS (or NHYP). Also, if IOLD≠0, then hyperplane coefficients and weights input here will be over-written by hyperplanes and weights from unit IOLD.

*This symbol indicates that no default is available. The variable must be supplied as an input by the user.

Table 4.4 ITPARM Namelist (Iteration Parameters)

Variable Name	Default Value	Definition
TIME1	*	If non-zero, the amount of time (in seconds) for P-function Initialization and Marginal Optimization
TIME2	*	Amount of time for Sequential Optimization
TIME3	*	Amount of time for Joint Optimization
FMIN(I) I=1,2,3	.001	Minimum RMS error in each of the three stages of fitting (used in gradient search scheme)
DFMIN(I) I=1,2,3	.001	Minimum change in RMS error allowed from one iteration to the next (used in gradient search scheme)
TKMIN(I) I=1,2,3	.001	Minimum step size (used in gradient search scheme)
GRADMIN(I) I=1,2,3	.001	Minimum size of gradient (used in gradient search scheme)
NSMX	5	Number of times in a row that the RMS error decreases before step size is increased (used in gradient search scheme)
TK	0.1	Initial value of step size in each type of iteration
TKFACT	4.0	Step size adjustment factor. This is the factor by which TK is divided (when RMS error increases) or multiplied (if NSMX decreases of RMS error in a row occur) (used in gradient search scheme).

*This symbol indicates that no default is available. The variable must be supplied as an input by the user.

Table 4.5 OTPARM Namelist (Output Parameters)

Variable Name	Default Value	Definition	
IPRINT	0	IPRINT value	Material Printed
		0	Input namelist, data standardization parameters, final P-functions for Marginal Optimization, and initial and final P-functions for Sequential and Joint Optimization. The output of P-functions (initial and final) show RMS error and percent variance explained, both biased and unbiased; also included are messages indicating on which of the five error criteria each iteration of the gradient search has terminated. The output ends with Table 1: Final P-functions.
		+1	Same as when IPRINT = 0 with the addition of Tables 2 through 7 which describe the resulting statistics from different points of view. See the annotation of the example in Section 5 for further details.
		-1	Same as +1 with a listing of all input/output data points, the predicted output variable, and the error at that point.
		+2	Same as +1 with the addition of the following: iteration termination parameter limits for each type of Optimization; initial P-functions (with RMS error and percent variance explained (both biased and unbiased) during P-function Initializations and Marginal Optimization; P-functions and weights being optimized every IPT iterations in the gradient search scheme; values of iteration termination criteria every IPT iterations.

Table 4.5 (Continued)

Variable Name	Default Value	Definition	Material Printed
		-2	Same as +2 with listing of data points (as in IPRINT = -1).
		+3	Same as +2 with the addition of the following: Iteration termination criteria whenever step size increases or decreases; gradients of P-functions and weights every IPT iterations.
		-3	Same as +3 with listing of data points (as in IPRINT = -1).
ISAVE	0	When non-zero, the number of the unit on which P-functions are being saved (in binary format).	
IPT	0	Frequency counter for printing of iteration termination criteria ($\|IPRINT\| \geq 2$), P-functions ($\|IPRINT\| \geq 2$), and gradients ($\|IPRINT\| \geq 3$). See definition of IPRINT above.	

4.3 ERROR MESSAGES AND COMMON PROBLEMS ENCOUNTERED

This section describes some common errors that can occur in the use of COMPLIAR and techniques for fixing these errors. Table 4.6 shows the possible error messages, the subroutine from which each is printed, and a brief statement of how to fix the error.

The error message in OPTW requires further explanation. It arises if the matrix $\underline{G}^T\underline{G}$, using the notation of Section 2.2.3, is singular.

The matrix $\underline{G}$ has elements $g_{\ell i}$ where, repeating Equation (2-20),

$$g_{\ell i} = P_i(\underline{x}^{\ell}) \quad , \quad g_{\ell,n+1} = 1 \qquad (4\text{-}1)$$

The explicit dependence of $g_{\ell i}$ on *only* $\underline{A}_i$ amongst all the hyperplane coefficients is suppressed for the moment to emphasize the structure of $\underline{G}$. The matrix $\underline{G}^T\underline{G}$ is singular whenever the number of *linearly independent* rows or columns of $\underline{G}$ is smaller than (N_p+1). This can occur, for example, if $M>(N_p+1)$ (the usual case) and if any two P-functions are identical. When this happens, it is clear that two columns of $\underline{G}$ will be identical (linearly dependent) which makes $\underline{G}^T\underline{G}$ singular.

In practice, there are two common situations which can cause two or more P-functions to be identical when OPTW is called:

(1) If two or more P-functions are input through the STPARM Parameter List with identical hyperplane coefficients

(2) In the course of P-function Initialization and Marginal Optimization if too little time is allotted (i.e., TIME1 is too small) then two P-functions may simultaneously:

Table 4.6 Error Messages

Error Message Output	Subroutine(s)	What to do
DIMENSION CONFLICT	CMPLAR	Change the number of input variables selected to match those on previous runs. This message can only appear when running from previously determined P-functions read out and back in binary format (IOLD≠0)
XXXX WORDS NEEDED FOR AAA ARRAY, ONLY XXXX WORDS AVAILABLE	CMPLAR SMNPUT	Allot more memory
NPFUNC MUST BE LESS THAN OR EQUAL TO 9. IT IS NOW XXXX	SMNPUT	Set NPFUNC between 1 and 9
NUMBER OF HYPERPLANES IN P-FUNCTION NUMBER N IS MM. IT MUST BE BETWEEN 1 AND 10	SMNPUT	Set all NHYP(I), I=1,...,NPFUNC between 1 and 10 (and NHYPS = 0) or set NHYPS between 1 and 10
OPTW FAILS, MATRIX SINGULAR	OPTW	See discussion in text of Section 4.3

(a) have the same values as originally initialized (i.e., no Marginal Optimization actually takes place)

(b) have the same number of hyperplanes

(c) be initialized on the same input variable (requires $N_p > n$).

The first situation above can be fixed by simply changing one of the P-functions a little.

The second situation will arise whenever TIME1 is sufficiently small and two P-functions that are initialized on the same variable have the same number of hyperplanes. This will happen when there are more P-functions than input variables. In this situation, if P_i and P_j are two P-functions, then the initial value of them will be the same if $i \equiv j \pmod{n}$, where n is the number of input variables, and $Q_i = Q_j$ (NHYP(i) = NHYP(j)).

Another possible situation which can cause $\underline{G}^T\underline{G}$ to be singular is if any of the P-functions are constant. This will cause two columns of $\underline{G}$ to be linearly dependent because of the column of "ones" present in $\underline{G}$. The fix is obviously to remove these trivial P-functions which have zeros for the coefficients of the input variables. The required constant will be absorbed by one of the remaining P-functions.

4.4 PRINTED OUTPUT OF THE PROGRAM

This section of the report describes the printed output produced by COMPLIAR. The major contribution to the output consists of seven printed tables (referred to below as Tables 1 through 7 in accordance with the example in Section 5). The reader will find it helpful to read the text of this section while glancing at the example output contained in Section 5 of this report.

The printed output of the program can be readily divided into four parts:

(1) A listing of the five parameter lists in which the user specifies how the program is to operate on the data

(2) Print from each of the phases of optimization used, where the user selects how much print will be produced through the absolute value of IPRINT (See Table 4.5)

(3) The series of seven tables in which overall statistics about the fit and detailed region statistics of the fit are displayed (the printing of these seven tables occurs when $|IPRINT| \geq 1$. Table 1 will be produced on all runs, however.

(4) A listing of all the data points used with the following information displayed for each point:

a) input variable
b) the output variable
c) the predicted output variable*
d) the error

(The printing of item (4) occurs after Table 1 and occurs when IPRINT is negative.)

An examination of the beginning of the run in Section 5 and Tables 4.1 through 4.5 shows clearly the first of the four parts of the output.

*This is the value of the output variable which is produced by the final repro-model.

The second phase of the output, that which occurs during optimization of the P-functions, is also exemplified by the portion of the run in Section 5 in which optimization occurs. The variables which control the amount of output during optimization are IPRINT and IPT. The elements of the output are decribed in Table 4.2.5 in the definitions of IPRINT and IPT. These outputs are shown and labeled (i.e., which outputs occur for which values of IPRINT, since IPT is just a frequency designator for some of the print) at the beginning of the P-function Initialization and Optimization print in the run in Section 5.

The main purpose of these optimization prints is to give the user some indication of what is actually going on in the optimization process. It can help to answer questions like: are things improving at a fast enough rate so that on a subsequent run more time should be allotted? How does adding more P-functions or otherwise changing the P-function structure affect the quality of the fit and the computational resources required?

The third type of print, the series of seven tables giving detailed statistics, is the most important output for analyzing the fit obtained. An examination of Tables 1 through 7 in the example of Section 5 is recommended along with a reading of the following description.

Table 1, which is divided into Tables 1A and 1B, displays the final fit produced by the program: the P-functions appropriate to the unstandardized (i.e., "original" input) data. Table 1 is printed after the optimization is finished, hyperplanes of insufficient activity are removed

(see Section 3.1.5) and the coefficients of the hyperplanes are transformed back to the original (i.e., unstandardized) coordinates. In Table 1A, the magnitude of the P-function weight has been absorbed into the hyperplane coefficients;* whereas in Table 1B the separate identity of the hyperplane coefficients and P-function weights has been preserved. Table 1A is a more economical presentation of the final fit. Table 1B is included in case the user wishes to input these P-functions, or a modification of them through the STPARM Parameter List.

The detailed contents of Table 1A are as follows: FINAL ERROR is the value of the overall RMS error obtained with the fit, and FINAL PCTVE is the corresponding per-cent variation explained.

The reader will note the terminology "biased" and "unbiased" appended to the numerical ERROR and PCTVE results. This terminology is borrowed from the general problem of linear regression as discussed, for example, in [3]. Thus, if one considers the expression for E in (2-15) as an estimate of the variance of the residuals, E is a biased estimate of this variance. The bias can be removed by altering the constant factor used in (2-15). In particular, for a regression fit composed of N_F free parameters, the unbiased estimate of the variance of the residuals is obtained by using the divisor $(M-N_F)$ in (2-15) instead of M.

Thus, when (2-15) is used as is, the resulting error is referred to as the BIASED MSE, and when the divisor $(M-N_F)$ is used instead of M, we refer

---

*Remember that the P-function weights may be either positive or negative. It is convenient to absorb the magnitude of the weights into the P-function specification and separately keep track of the algebraic sign. The way to combine this information to produce the final repro-model is illustrated in the example. (See Table 1A on page 94.)

[3] Draper, N. R. and H. Smith, Applied Regression Analysis, Wiley & Sons, 1966, p. 58-63.

to the resulting error as the UNBIASED MSE. The RMS errors are obtained by taking the square root of the appropriate MSE. Similarly, the appropriate value of PCTVE is obtained using the biased or unbiased estimator of the MSE.* PCTVE is evaluated from

$$\text{PCTVE} = \left[1 - \frac{\sigma_F^2}{\sigma_y^2} \right] \times 100\%$$

where σ_F^2 is the biased or unbiased estimate of the MSE and σ_y^2 is the variance of the dependent (output) variable calculated from the data supplied.

Several comments are now in order regarding the interpretation of the results. Generally, the biased RMS error and PCTVE results are satisfactory for judging the quality of the fit. In any case, if $M \gg N_F$ there will be very little difference in the numerical results anyway (though the biased RMS error will always be the smaller and thereby indicate a higher PCTVE). When M is not much larger than N_F there is a danger that too many degrees of freedom are being utilized in the fit for the amount of data available. After all, with any finite set of data points, if a sufficient number of parameters are allowed in the fit, the data can be fit exactly. This is "goldmining" and does not necessarily result in a good fit over the overall input/output variable space. In other words, the real use of the biased and unbiased numerical results is to indicate when goldmining or overfitting may be occurring.

*Unless otherwise noted in the printed output, all RMS errors and PCTVE values are obtained using the more common BIASED MSE expression.

If these numerical results are approximately equal, then M is sufficiently large for the complexity of the fit attempted. A large discrepancy between the biased and unbiased numerical results is a strong indication that either a less complicated fit should be used, or more data must be utilized in the fit.

As a further safeguard, if $M \leq N_F$ (in which case the unbiased MSE would produce a negative result!) the unbiased RMS error output is set to +999 and the unbiased PCTVE is set to -999. Also, an error message is printed warning of the paucity of data.

Finally, we mention that there are situations which may cause PCTVE to be negative. This occurs if the quality of the fit is poor. Instead of printing the negative (and meaningless) PCTVE results, they are redefined to be zero to make it easy to identify.

Table 2, "Overall Statistics," provides the mean and standard deviation of all n input variables, the output variable, and the predicted variable. Also the full covaraiance matrix and correlation matrix are provided for the (n+2)-dimensional vector composed of: the n input variables; the single output variable; and the value of the output variable predicted by the fit. These matrices indicate whether or not the correlations between each of the input variables and the output variables is similar to the correlation between each of the input variables and the predicted variables. Also, these matrices reveal the correlation between the output variable and the predicted variable.

Table 3, "Net Hyperplane Statistics for XXX Regions," gives a listing of regions in decreasing order of population. The following information is supplied for each region: the region I.D., the population, the percentage of the total population, the RMS error (taken over the points in each region separately), and the percent variation explained in that region. While most of these terms are self explanatory, the region I.D. is not. This item is explained by the note provided on the sample output of Table 3 in Section 5.

Table 4, "Qualitative Characterization of the Subregions," is designed to give some overall indication of how, in each region, the mean of the variable compares to the overall mean. The unit of comparison is the standard deviation of the particular variable. The way the numbers in Table 4 are arrived at is as follows: Call the mean of the $k^{\underline{th}}$ variable in the $j^{\underline{th}}$ region μ_{kj} and let μ_k be the mean of the $k^{\underline{th}}$ variable over all regions. Also, let σ_k represent the standard deviation of the $k^{\underline{th}}$ variable overall. Then let

$$R_{kj} = \frac{\mu_{kj} - \mu_k}{\sigma_k}$$

The quantity R_{kj} is a measure of how many standard deviations the mean, μ_{kj} of the $k^{\underline{th}}$ variable in region j, lies from the global mean, μ_k. The value printed in Table 4 is derived from R_{kj} is as follows:

Value of R_{kj}	Value Printed for Region j and Variable k
$R_{kj} \leq -1$	-3
$-1 \leq R_{kj} \leq -\frac{1}{2}$	-2
$-\frac{1}{2} \leq R_{kj} < 0$	-1
$0 \leq R_{kj} \leq \frac{1}{2}$	1
$\frac{1}{2} \leq R_{kj} \leq 1$	2
$1 \leq R_{kj}$	3

Note that the regions are numbered the same in Tables 3 through 7. Note also, that the variables on which the calculation of R_{kj} is done, include the output variable and the predicted variable.

Table 6, "Parameters of Fit for Actual (Non-Standardized) Data," lists the resultant hyperplane coefficients appropriate for the nonstandardized data. This table provides the actual equation, in each subregion, between the input and output variables.

Table 7, "Parameters of Fit for Standardized Independent Variables," lists the resultant hyperplane coefficients appropriate to the standardized input variables. This display is most useful for guaging the relative importance of the input variables in each region, since these are the coefficients of dimensionless variables.

4.5 OTHER OUTPUT OF THE PROGRAM

The program also outputs P-functions in binary format when ISAVE is nonzero, and it outputs them to unit ISAVE. As can be readily seen from inspecting the listing of CMPLAR, the following are output in the order described

NDIM1, NHYPT, NPFUNC, NHYP(I), I=1,...,NPFUNC

hyperplane coefficients

weights

V. SOME EXAMPLES

This section of the report is intended to furnish greater insight into the repro-modeling concept in general, and the COMPLIAR software in particular.

A sample run of COMPLIAR for a 4-dimensional case is presented in detail in Section 5.1. This run furnishes a repro-model which explains over 95% of the variation between the input and output variables.

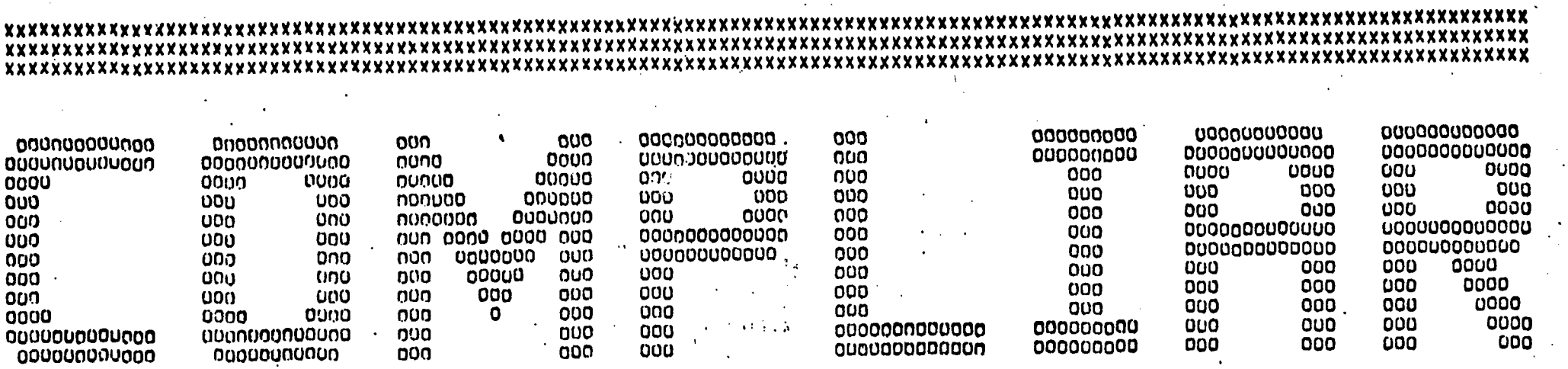
Section 5.2 describes the results of another run of COMPLIAR for an 8-dimensional case. The simplicity of the repro-model, even in this difficult case, is a striking example of the benefits of repro-modeling.

5.1 A SAMPLE RUN

Following is a sample COMPLIAR run. This run utilized 400 data points to fit the dependence of vehicle velocity (the "output" variable) upon four terrain parameters (the "input" variables). First, in Table 5.1, is shown a listing of the input cards actually used to generate this run. Following this table is the run itself. The only editing that has been done in presenting the results is to delete some of the iteration print in the optimization section to save space. These deletions are indicated where they occur.

Table 5.1 Listing of Input Cards

```
$INPARM  NSAMP=400  $
$STPARM  NPFUNC=2, NHYPS=2  $
$ITPARM  TIME1=10.,, TIME2=10.,, TIME3=10.,, DFMIN=2,E=4  $
$OTPARM  IPRINT=-3  $
$DDPARM  NVAR=9, NOFS=4, LOC=6,5,7,8, LOCDEP=9, JVAR=3HIGR,4HIRCI,3HIPR,3HIOH,
         JVARD=3HVEL  $
(5E15.6,/,4E15.6)
```



CONTINUOUS MULTIVARIATE PIECEWISE-LINEAR APPROXIMATION / REGRESSION

DEVELOPED BY: TECHNOLOGY SERVICE CORPORATION
2811 WILSHIRE BOULEVARD
SANTA MONICA, CALIFORNIA 90403

FOR: U.S. ARMY TANK AUTOMOTIVE RESEARCH
AND DEVELOPMENT COMMAND (TARADCOM)
WARREN, MICHIGAN

********** INPUT DATA DEFINITION ********** (DDPARM NAMELIST) **********

NUMBER OF INPUT VARIABLES = 9

INPUT FORMAT = BINARY

INDEPENDENT VARIABLE NUMBER 1 = IGR LOCATION = 6

INDEPENDENT VARIABLE NUMBER 2 = IRCI LOCATION = 5

INDEPENDENT VARIABLE NUMBER 3 = IPR LOCATION = 7

INDEPENDENT VARIABLE NUMBER 4 = IOH LOCATION = 8

DEPENDENT VARIABLE = VEL LOCATION = 9

This resulted from the inputs:

NVAR = 9
NOFS = 4
LØC = 6,5,7,8
LØCDEP = 9
JVAR = 3HIGR, 4HIRCI, 3HIPR, 3HIØH
JVARD = 3HVEL

********** INPUT PARAMETERS ********** (INPARM NAMELIST) **********

NSAMP = 400 MAXIMUM NUMBER OF SAMPLES ACCEPTED

IVERY = 1 WHEN EQUAL TO N, CAUSES EVERY NTH SAMPLE POINT TO BE READ (NOMINALLY = 1)

IUNIT = 1 INPUT UNIT FROM WHICH SAMPLE POINTS ARE READ

IOLD = 0 WHEN NON-ZERO, P-FUNCTIONS ARE READ FROM THIS UNIT (NOMINALLY 0)

********** STRUCTURE PARAMETERS ********** (STPARM NAMELIST) **********

NPFUNC = 2 NO. OF P-FUNCTIONS (1-9). 0 = CORRELATION MATRIX ONLY, -1 =SAMPLES LISTED ALSO

NHYPS = 2 NO. OF HYPERPLANES PER P-FUNCTION. IF = 0, NHYP VECTOR CONTAINS NUMBER OF HYPS. FOR EACH P-FUNCTION

ACTMIN = 1 MINIMUM NO. OF DATA POINTS PER HYPERPLANE

********** ITERATION PARAMETERS ********** (ITPARM NAMELIST) **********

TIME1 = 10 TIME (IN SECONDS) ALLOWED FOR P-FUNCTION INITIALIZATION AND MARGINAL OPTIMIZATION

TIME2 = 10 TIME ALLOWED FOR SEQUENTIAL OPTIMIZATION

TIME3 = 10 TIME ALLOWED FOR JOINT OPTIMIZATION

FMIN(1) = 1.00E-03 MINIMUM RMS ERROR IN 3 STAGES OF FITTING
FMIN(2) = 1.00E-03
FMIN(3) = 1.00E-03

DFMIN(1) = 2.00E-04 MINIMUM ERROR CHANGE IN 3 STAGES OF FITTING,
DFMIN(2) = 2.00E-04
DFMIN(3) = 2.00E-04

TKMIN(1) = 1.00E-03 MINIMUM STEP SIZE IN 3 STAGES OF FITTING
TKMIN(2) = 1.00E-03
TKMIN(3) = 1.00E-03

GRADMN(1) = 1.00E-03 MINIMUM MAGNITUDE OF GRADIENT IN 3 STAGES OF FITTING
GRADMN(2) = 1.00E-03
GRADMN(3) = 1.00E-03

NSMX = 5 NUMBER OF SUCCESSFUL MOVES BEFORE STEP SIZE IS INCREASED

TK = 1.00E-01 INITIAL STEP SIZE IN EACH STEP OF FITTING

TKFACT = 4.00 FACTOR BY WHICH STEP SIZE IS MULTIPLIED (OR DIVIDED) WHEN NSMX SUCCESSES (OR 1 FAILURE) OCCUR

********** OUTPUT PARAMETERS ********** (OTPARM NAMELIST) **********

IPRINT = -3 PRINT FLAG

VALUE	RESULTING OUTPUT
0	MINIMUM, INCLUDING INPUT PARAMETERS, FINAL P-FUNCTIONS, COMPUTING TIME
+1	SAME AS 0 WITH NET HYPERPLANE STATISTICS
+2	SAME AS 1 WITH DIAGNOSTIC PRINT DURING OPTIMIZATION
+3	SAME AS 2 WITH FULL DIAGNOSTIC PRINT DURING OPTIMIZATION
-1,-2,-3	SAME AS +1,2,3 WITH ALL DATA POINTS AND FITTED VALUES PRINTED AFTER FITTING

ISAVE = 30 IF NON-ZERO, NO. OF UNIT ON WHICH P-FUNCTIONS ARE SAVED

IPT = 1 WHEN = N, PRINTING OF P-FUNCTIONS (IPRINT=2) OR P-FUNCTIONS AND GRADIENT (IPRINT=3) OCCURS EACH NTH ITERATION

DATA STANDARDIZATION PARAMETERS FOR 400 POINTS **********

	IGR	IRCI	IPR	IOH	VEL
MEAN	4.000	5.500	3.000	2.500	9.969
S.D.	1.414	1.118	1.414	1.118	4.643

```
*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/
P-FUNCTION INITIALIZATION AND MARGINAL OPTIMIZATION ******************************** TIME =    5.49 ***********************************
*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/

ITERATION TERMINATION PARAMETER LIMITS  --------------------

TIME ALLOTTED =  10.0   RMS ERROR =  1.000E-03   ERROR CHANGE =  2.000E-04   STEP SIZE = 1.000E-03   GRADIENT MAGNITUDE =  1.000E-03
                                                  --------------------
```

This print always occurs.
Iteration parameter limits printed for |IPRINT|≥2
These are the limits against which each of these five parameters will be tested (see notes at terminations on time criterion)

```
INITIAL DETERMINATION OF P-FUNCTION NUMBER  1 ***********************

P-FUNCTION NO.  1

        IGR            IRCI          IFR           IOH           CONST

    -1.000E+00    0.            0.            0.            0.
     1.000E+00    0.            0.            0.            0.

    WEIGHT = -1.415E+00

   CONSTANT TERM =  1.117E+01

   INITIAL ERROR =   4.58     (BIASED)        4.65     (UNBIASED)

   INITIAL PCTVE =   2.60     (BIASED)          0.     (UNBIASED)
```

This print occurs when |IPRINT|≥2

```
ITER NO.=   0  TIME=   6.34  RMS ERR=   4.582      ERR CHG=  10000.      STPSIZ=  .1000  GRADMAG=   634.0       NO. OF SUCCESSES=   0

P-FUNCTION NO.  1

    -1.000E+00    0.            0.            0.            0.
     1.000E+00    0.            0.            0.            0.

      WEIGHT = -1.415E+00

   CONSTANT TERM =  1.117E+01
```

This print of the above summary line and of the P-function being optimized occurs every IPT iterations whenever |IPRINT|≥2

```
GRADIENT OF NO.  1

    -1.072E+03  -7.463E+02  -5.480E+02  -8.582E+01   8.963E+02
    -1.072E+03  -2.174E+02  -7.605E+01  -9.561E+00  -8.963E+02

      DWEIGHT =  1.072E-09

  DCONSTANT TERM = -2.139E-09
```

This print of the gradient of the P-function being optimized occurs every IPT iterations whenever |IPRINT|=3

```
ITER NO.=   1  TIME=   6.69  RMS ERR=   4.182      ERR CHG=   .3999      STPSIZ=  .1000  GRADMAG=   573.9      NO. OF SUCCESSES=   1
 P-FUNCTION NO.  1

     -8.309E-01   1.177E-01   8.645E-02   1.354E-02  -1.414E-01
      1.169E+00   3.430E-02   1.200E-02   1.508E-03   1.414E-01

       WEIGHT =  -1.415E+00

   CONSTANT TERM =  1.117E+01

GRADIENT OF NO.  1

     -9.564E+02  -5.146E+02  -3.966E+02  -6.291E+01   8.123E+02
     -9.564E+02  -3.950E+02  -1.940E+02  -2.741E+01  -7.897E+02

       DWEIGHT =  4.676E+02

  DCONSTANT TERM = -1.600E+01
```

The P-function and gradient print for iterations 2-10 has been omitted.

```
   STEP SIZE INCREASE - ITER NO.=   5  TIME=   7.70  RMS ERR=   2.310      ERR CHG=   .4778      STPSIZ=  .4000  GRADMAG=   460.8

   STEP SIZE DECREASE - ITER NO.=   6  TIME=   8.04  RMS ERR=   2.310      ERR CHG=   .4778      STPSIZ=  .1000  GRADMAG=   351.7

   STEP SIZE DECREASE - ITER NO.=  10  TIME=   9.41  RMS ERR=   1.425      ERR CHG=   .1670      STPSIZ=  .0250  GRADMAG=   68.06
```

```
ITER NO.=  11  TIME=  10.10  RMS ERR=   1.398      ERR CHG=  2.7209E-02  STPSIZ=  .0250  GRADMAG=   41.85      NO. OF SUCCESSES=   9
P-FUNCTION NO.  1

      1.906E-01   8.831E-01   8.028E-01   1.175E-01  -8.649E-01
      1.983E+00   5.982E-01   2.454E-01   4.158E-02   2.703E-01

      WEIGHT =  -2.453E+00

   CONSTANT TERM =  1.146E+01

GRADIENT OF NO.  1

     -1.912E+00  -5.991E+01  -8.792E+01   7.566E+00  -4.010E+01
     -9.075E+00   8.548E+01   2.591E+01  -1.196E+01   6.627E+01

      DWEIGHT =  2.258E+01

  DCONSTANT TERM =  -1.067E+01
```

```
ITER NO.=  12  TIME=  10.44  RMS ERR=   1.382     ERR CHG=  1.5958E-02  STPSIZ=  .0250  GRADMAG=   21.67     NO. OF SUCCESSES=  10
P-FUNCTION NO.  1

      1.918E-01   9.189E-01   8.554E-01   1.130E-01  -8.409E-01
      1.989E+00   5.710E-01   2.299E-01   4.992E-02   2.307E-01

      WEIGHT = -2.467E+00

   CONSTANT TERM =  1.147E+01

GRADIENT OF NO.  1

     -2.496E+01  -2.471E+01  -3.973E+01   2.561E+00  -1.116E+01
     -2.570E+01   1.285E+01   1.526E+01   3.344E+00   1.871E+01

      DWEIGHT =  3.551E+01

  DCONSTANT TERM = -3.061E+00

****************************** ITERATION TERMINATED ON TIME
ITER NO.=  13  TIME=  10.78  RMS ERR=   1.378     ERR CHG=  3.2423E-03  STPSIZ=  .0250  GRADMAG=   30.85     NO. OF SUCCESSES=  11

   FINAL ERROR =   1.38     (BIASED)      1.40     (UNBIASED)
   FINAL PCTVE =   91.2     (BIASED)      90.9     (UNBIASED)
   ACTIVITY VECTOR =      136     264
P-FUNCTION NO.  1
        IGR         IRCI        IPR         IOH         CONST
      2.206E-01   9.474E-01   9.012E-01   1.100E-01  -8.281E-01
      2.018E+00   5.562E-01   2.125E-01   4.606E-02   2.091E-01

     WEIGHT = -2.508E+00

   CONSTANT TERM =  1.147E+01

TIME =   11.12

   CUMULATIVE ERROR =   1.38     (BIASED)      1.40     (UNBIASED)
   CUMULATIVE PCTVE =   91.2     (BIASED)      91.0     (UNBIASED)
```

Time allotted for each P-function in this stage of optimization is TIME1/NPFUNC, which is 5 sec in this case. Hence, the limit is 5.49 + 5 = 10.49 sec.

This line is printed on termination when |IPRINT| ≥ 2.

These two blocks of print will always occur.

This computation of the error and percent variation explained is performed after a final optimization of the weights.

```
INITIAL DETERMINATION OF P-FUNCTION NUMBER  2 **********************

P-FUNCTION NO.  2

      IGR          IRCI          IRR          IOH          CONST

     0.          -1.000E+00    0.           0.           0.
     0.           1.000E+00    0.           0.           0.

   WEIGHT = -6.132E-01

  CONSTANT TERM =  5.486E-01


    INITIAL ERROR =   1.35     (BIASED)        1.37     (UNBIASED)

    INITIAL PCTVE =   91.6     (BIASED)        91.3     (UNBIASED)

ITER NO.=   0  TIME=  12.65  RMS ERR=   1.349      ERR CHG=  10000.      STPSIZ=  .1000  GRADMAG=   45.60      NO. OF SUCCESSES=  0

P-FUNCTION NO.  2

      0.          -1.000E+00    0.           0.           0.
      0.           1.000E+00    0.           0.           0.

       WEIGHT = -6.132E-01

   CONSTANT TERM =  5.485E-01


GRADIENT OF NO.  2

     -4.188E+01   -3.459E-01   -1.042E+02   -9.292E+00    1.732E+00
      3.860E+01   -3.459E-01    1.034E+02    8.973E+00   -1.732E+00

       DWEIGHT =  5.869E-11

   DCONSTANT TERM = -1.396E-10
```

P-function and gradient print for iterations 1-8 has been omitted.

```
STEP SIZE DECREASE - ITER NO.=  5  TIME=  13.99  RMS ERR=   1.002      ERR CHG=  5.9195E-02  STPSIZ=  .0250  GRADMAG=   47.11
```

********** ITERATION TERMINATED ON TIME

Time limit for this P-function is 5.49 + 10 = 15.49 sec.

ITER NO.= 9 TIME= 15.71 RMS ERR= .9364 ERR CHG= 1.0873E-02 STPSIZ= .0250 GRADMAG= 12.13 NO. OF SUCCESSES= 8

FINAL ERROR = .936 (BIASED) .950 (UNBIASED)

FINAL PCTVE = 95.9 (BIASED) 95.8 (UNBIASED)

ACTIVITY VECTOR = 204 196

P-FUNCTION NO. 2

IGR	IRCI	IPR	IOH	CONST
2.177E-01	-9.190E-01	8.234E-01	1.048E-02	-1.501E-01
-2.011E-01	9.548E-01	-8.411E-01	-8.832E-03	-2.464E-01

WEIGHT = -1.197E+00

CONSTANT TERM = 9.629E-01

TIME = 16.05

CUMULATIVE ERROR = .924 (BIASED) .950 (UNBIASED)

CUMULATIVE PCTVE = 96.0 (BIASED) 95.8 (UNBIASED)

```
*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/
SEQUENTIAL OPTIMIZATION ************************************************************ TIME =   17.44 *********************************
*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/


ITERATION TERMINATION PARAMETER LIMITS  -------------------

TIME ALLOTTED =  10.0   RMS ERROR =  1.000E-03   ERROR CHANGE =  2.000E-04   STEP SIZE = 1.000E-03   GRADIENT MAGNITUDE =  1.000E-03
                                                 -------------------


INITIAL P-FUNCTIONS

P-FUNCTION NO.  1

        IGR          IRCI          IPR          IOH          CONST

     2.206E-01    9.474E-01    9.012E-01    1.100E-01   -8.281E-01
     2.018E+00    5.562E-01    2.123E-01    4.606E-02    2.091E-01

     WEIGHT = -2.472E+00

P-FUNCTION NO.  2

        IGR          IRCI          IPR          IOH          CONST

     2.177E-01   -9.190E-01    8.234E-01    1.048E-02   -1.501E-01
    -2.011E-01    9.548E-01   -8.411E-01   -8.832E-03   -2.464E-01

     WEIGHT = -1.398F+00

    CONSTANT TERM =  1.265E+01

    INITIAL ERROR =   .924    (BIASED)        .950    (UNBIASED)

    INITIAL PCTVE =   96.0    (BIASED)        95.8    (UNBIASED)

          ITERATION NUMBER   1  ************************ TIME =   18.13 ************************************************************

                   ITERATE ON P-FUNCTION NUMBER  1  ************************************************

ITER NO.=   0  TIME=  19.00  RMS ERR=   .9237      ERR CHG=  10000.      STPSIZ=  .1000  GRADMAG=   38.02     NO. OF SUCCESSES=   0

P-FUNCTION NO.  1

     2.206E-01    9.474E-01    9.012E-01    1.100E-01   -8.281E-01
     2.018E+00    5.562E-01    2.123E-01    4.606E-02    2.091E-01

     WEIGHT = -2.472E+00

    CONSTANT TERM =  6.562E-12
```

The iteration number here refers to the big loop of Sequential Optimization in which all of the P-functions are optimized (if there is enough time).

```
GRADIENT OF NO,  1

      6,407E+00  -1,173E+01  -6,698E+01  -1,177E+00  -5,192E+01
      1,459E+00  -8,447E+00   8,493E+01   1,283E+00   5,192E+01

      DWEIGHT =  2,157E-10

   DCONSTANT TERM =  2,234E-11


  STEP SIZE DECREASE - ITER NO,=   1  TIME=  19,00  RMS ERR=   ,9237      ERR CHG=  10000,      STPSIZ=  ,0250  GRADMAG=   38,02

ITER NO,=   1  TIME=  19,33  RMS ERR=   ,9237      ERR CHG=  10000,      STPSIZ=  ,0250  GRADMAG=   38,02      NO, OF SUCCESSES=   0

P-FUNCTION NO,  1

      2,206E-01   9,474E-01   9,012E-01   1,100E-01  -8,281E-01
      2,018E+00   5,562E-01   2,123E-01   4,606E-02   2,091E-01

      WEIGHT =  -2,472E+00

    CONSTANT TERM =  6,562E-12

GRADIENT OF NO,  1

      6,407E+00  -1,173E+01  -6,698E+01  -1,177E+00  -5,192E+01
      1,459E+00  -8,447E+00   8,493E+01   1,283E+00   5,192E+01

      DWEIGHT =  2,107E-10

   DCONSTANT TERM =  2,234E-11
```

87

```
STEP SIZE DECREASE - ITER NO,=   4  TIME=  19,99  RMS ERR=   ,8994      ERR CHG=  5,5564E-03  STPSIZ=  ,0063  GRADMAG=   34,08
```

P-function and gradient print for iterations 2-7 have been omitted.

```
********************************* ITERATION TERMINATED ON TIME

ITER NO.=   8  TIME=  21.65  RMS ERR=   .8887      ERR CHG=  1.6046E-03  STPSIZ=  .0063  GRADMAG=   10.69      NO. OF SUCCESSES=   6

P-FUNCTION NO.  1

        IGR.          IRCI          IPR           IOH          CONST

      2.795E-01    9.482E-01    9.733E-01    1.004E-01   -7.324E-01
      2.061E+00    5.406E-01    1.265E-01    4.893E-02    1.140E-01

    WEIGHT = -2.492E+00

  CONSTANT TERM = -2.427E-04
```

This is still within the first big loop of Sequential Optimization.

```
                ITERATE ON P-FUNCTION NUMBER  2 ***********************************************

ITER NO.=   0  TIME=  22.54  RMS ERR=   .8875      ERR CHG=  10000.      STPSIZ=  .1000  GRADMAG=   12.33      NO. OF SUCCESSES=   0


P-FUNCTION NO.  2

      2.177E-01   -9.190E-01    8.234E-01    1.048E-02   -1.501E-01
     -2.011E-01    9.548E-01   -8.411E-01   -8.832E-03    2.464E-01

      WEIGHT = -1.454E+00

  CONSTANT TERM =  2.548E-02

GRADIENT OF NO.  2

     -2.775E+01   -1.240E+01   -6.636E+00    5.368E+00   -1.518E+01
      7.334E+00    1.283E+01    1.107E+01   -5.151E+00    1.518E+01

      DWEIGHT =  3.020E-11

  DCONSTANT TERM = -3.661E-11
```

P-function and gradient print for iterations 1-7 have been omitted.

```
 STEP SIZE DECREASE = ITER NO.=   5  TIME=  23.86  RMS ERR=   .8759      ERR CHG=  1.6984E-03  STPSIZ=  .0063  GRADMAG=   5.891

 STEP SIZE DECREASE = ITER NO.=   8  TIME=  24.85  RMS ERR=   .8747      ERR CHG=  3.5849E-04  STPSIZ=  .0016  GRADMAG=   3.195
********************************** ITERATION TERMINATED ON TIME
ITER NO.=   8  TIME=  25.18  RMS ERR=   .8747      ERR CHG=  3.5849E-04  STPSIZ=  .0016  GRADMAG=   3.195      NO. OF SUCCESSES=   5
P-FUNCTION NO.  2
        IGR          IRCI         IPR          IOH          CONST

      2.843E-01   -8.251E-01    7.826E-01   -5.502E-03   -5.248E-02
     -2.584E-01    9.762E-01   -9.161E-01    7.717E-03   -3.593E-01

     WEIGHT = -1.463E+00

    CONSTANT TERM =  3.596E-02

SEQUENTIAL OPTIMIZATION TERMINATES ==========

    FINAL ERROR =    .874      (BIASED)        .900      (UNBIASED)

    FINAL PCTVE =   96.5       (BIASED)       96.2       (UNBIASED)

    ACTIVITY VECTOR =       143      257      212      188

P-FUNCTION NO.  1
        IGR          IRCI         IPR          IOH          CONST

      2.795E-01    9.482E-01    9.733E-01    1.004E-01   -7.324E-01
      2.061E+00    5.406E-01    1.265E-01    4.893E-02    1.140E-01

     WEIGHT = -2.482E+00

P-FUNCTION NO.  2
        IGR          IRCI         IPR          IOH          CONST

      2.843E-01   -8.251E-01    7.826E-01   -5.502E-03   -5.248E-02
     -2.584E-01    9.762E-01   -9.161E-01    7.717E-03   -3.593E-01

     WEIGHT = -1.476E+00

    CONSTANT TERM =  1.268E+01
```

```
*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/
JOINT OPTIMIZATION ****************************************************** TIME =   26.24 ******************************************************
*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/


ITERATION TERMINATION PARAMETER LIMITS  --------------------
TIME ALLOTTED =  10.0   RMS ERROR =  1.000E-03   ERROR CHANGE =  2.000E-04   STEP SIZE = 1.000E-03   GRADIENT MAGNITUDE =  1.000E-03
                                          --------------------


P-FUNCTION NO.  1
        IGR          IRCI         IPR          IOH         CONST

      2.793E-01    9.482E-01    9.733E-01    1.004E-01   -7.324E-01
      2.061E+00    5.406E-01    1.265E-01    4.893E-02    1.140E-01

    WEIGHT = -2.482E+00
P-FUNCTION NO.  2
        IGR          IRCI         IPR          IOH         CONST

      2.843E-01   -8.251E-01    7.826E-01   -5.502E-03   -5.248E-02
     -2.584E-01    9.762E-01   -9.161E-01    7.717E-03   -3.593E-01

    WEIGHT = -1.476E+00

   CONSTANT TERM =  1.268E+01

   INITIAL ERROR =   .874      (BIASED)        .900      (UNBIASED)

   INITIAL PCTVE =   96.5      (BIASED)        96.2      (UNBIASED)

ITER NO.=   0  TIME=  27.47  RMS ERR=   .8744      ERR CHG=  10000.      STPSIZ=  .1000  GRADMAG=   12.00      NO. OF SUCCESSES=   0
P-FUNCTION NO.  1
      2.793E-01    9.482E-01    9.733E-01    1.004E-01   -7.324E-01
      2.061E+00    5.406E-01    1.265E-01    4.893E-02    1.140E-01

      WEIGHT = -2.482E+00
P-FUNCTION NO.  2
      2.843E-01   -8.251E-01    7.826E-01   -5.502E-03   -5.248E-02
     -2.584E-01    9.762E-01   -9.161E-01    7.717E-03   -3.593E-01

      WEIGHT = -1.476E+00

   CONSTANT TERM =  1.268E+01
```

```
 GRADIENT OF NO.  1

      -1.388E+01   9.029E+00  -2.271E+01  -3.491E+00  -2.638E+01
       1.855E+00  -2.131E+01   2.369E+01   2.798E+00   2.638E+01

       DWEIGHT =  3.499E-11

 GRADIENT OF NO.  2

      -3.993E+00  -4.132E+00   1.070E+00  -1.600E-01  -4.170E+00
      -3.156E+00  -3.167E+00  -4.854E-01  -2.520E-01   4.170E+00

       DWEIGHT =  5.555E-11

    DCONSTANT TERM = -2.331E-09

  STEP SIZE DECREASE - ITER NO.=   1  TIME=  27.47  RMS ERR=   .8744     ERR CHG=  10000.     STPSIZ=  .0250  GRADMAG=   12.00

ITER NO.=   1  TIME=  28.11  RMS ERR=   .8744     ERR CHG=  10000.     STPSIZ=  .0250  GRADMAG=   12.00     NO. OF SUCCESSES=   0

 P-FUNCTION NO.  1

       2.793E-01   9.482E-01   9.733E-01   1.004E-01  -7.324E-01
       2.061E+00   5.406E-01   1.265E-01   4.893E-02   1.140E-01

       WEIGHT = -2.482E+00

 P-FUNCTION NO.  2

       2.843E-01  -8.251E-01   7.826E-01  -5.512E-03  -5.248E-02
      -2.584E-01   9.762E-01  -9.161E-01   7.717E-03  -3.593E-01

       WEIGHT = -1.476E+00

    CONSTANT TERM =  1.268E+01

 GRADIENT OF NO.  1

      -1.388E+01   9.029E+00  -2.271E+01  -3.491E+00  -2.638E+01
       1.855E+00  -2.131E+01   2.369E+01   2.798E+00   2.638E+01

       DWEIGHT =  3.409E-11

 GRADIENT OF NO.  2

      -3.993E+00  -4.132E+00   1.070E+00  -1.600E-01  -4.170E+00
      -3.156E+00  -3.167E+00  -4.854E-01  -2.520E-01   4.170E+00

       DWEIGHT =  5.540E-11

    DCONSTANT TERM = -2.325E-09
```

Note gradient contains 23 terms at this stage of optimization.

P-function and gradient print for iterations 2-15 have been omitted.

STEP SIZE DECREASE - ITER NO.= 3 TIME= 28.73 RMS ERR= .8692 ERR CHG= 5.2399E-03 STPSIZ= .0063 GRADMAG= 23.41

STEP SIZE INCREASE - ITER NO.= 8 TIME= 31.86 RMS ERR= .8500 ERR CHG= 2.1061E-03 STPSIZ= .0250 GRADMAG= 10.60

STEP SIZE DECREASE - ITER NO.= 9 TIME= 32.48 RMS ERR= .8500 ERR CHG= 2.1061E-03 STPSIZ= .0063 GRADMAG= 13.36

STEP SIZE INCREASE - ITER NO.= 14 TIME= 35.59 RMS ERR= .8402 ERR CHG= 1.5717E-03 STPSIZ= .0250 GRADMAG= 12.96

STEP SIZE DECREASE - ITER NO.= 15 TIME= 36.22 RMS ERR= .8402 ERR CHG= 1.5717E-03 STPSIZ= .0063 GRADMAG= 13.34

```
    FINAL ERROR =    .839      (BIASED)       .863     (UNBIASED)

    FINAL PCTVE =   96.7       (BIASED)       96.5     (UNBIASED)

    ACTIVITY VECTOR =      162      238      221      179

P-FUNCTION NO.  1

        IGR          IRCI         IPR          IOH          CONST

      3.975E-01   9.007E-01   1.000E+00   9.957E-02  -5.815E-01
      2.082E+00   5.816E-01   5.567E-02   4.779E-02  -6.807E-04

     WEIGHT = -2.480E+00

P-FUNCTION NO.  2

        IGR          IRCI         IPR          IOH          CONST

      4.057E-01  -8.188E-01   7.661E-01  -1.054E-02  -3.406E-03
     -2.966E-01   9.620E-01  -9.259E-01   1.159E-02  -5.867E-01



     WEIGHT = -1.512E+00

    CONSTANT TERM =  1.265E+01

TIME =   37.88
```

FINAL WEIGHTED P-FUNCTIONS (STANDARDIZATION REMOVED, MAGNITUDE OF P-FUNCTION WEIGHTS ABSORBED) ********** TABLE 1A *********

FINAL ERROR = .839 (BIASED) .861 (UNBIASED)

FINAL PCTVE = 96.7 (BIASED) 96.6 (UNBIASED)

P-FUNCTION NO. 1 SIGN OF WEIGHT IS =

IGR	IRCI	IPR	IOH	HYPCONST
6.971E-01	1.998E+00	1.754E+00	2.209E-01	-2.103E+01
3.651E+00	1.290E+00	9.761E-02	1.060E-01	-2.226E+01

P-FUNCTION NO. 2 SIGN OF WEIGHT IS =

IGR	IRCI	IPR	IOH	HYPCONST
4.336E-01	-1.102E+00	8.191E-01	-1.426E-02	1.899E+00
-3.172E-01	1.301E+00	-9.899E-01	1.568E-02	-3.541E+00

CONSTANT TERM = 1.265E+01

The correct algebraic sign of the P-function weight is indicated here for use in the final repro-model.

Elaboration: Let $H_q^{(i)}$ denote the q^{th} hyperplane of the i^{th} P-function

P-function No. 1 consists of the following two hyperplanes

$$H_1^{(1)}(\underline{x}) = +.6971\ \text{IGR} + 1.998\ \text{IRCI} + 1.754\ \text{IPR} + .2209\ \text{IOH} - 21.03$$

$$H_2^{(1)}(\underline{x}) = +3.651\ \text{IGR} + 1.290\ \text{IRCI} + .09761\ \text{IPR} + .0106\ \text{IOH} - 22.26$$

where $\underline{x} = [x_1\ x_2\ x_3\ x_4] = [\text{IGR IRCI IPR IOH}]$

P-function No. 2 consists of the two hyperplanes $H_1^{(2)}(\underline{x})$ and $H_2^{(2)}(\underline{x})$ similarly obtained.

Using the information on the sign of the P-function weights, the final repro-model is:

$$\text{VEL}(\underline{x}) = (-1)\cdot\text{Max}\left\{H_1^{(1)}(x), H_2^{(1)}(x)\right\} + (-1)\cdot\text{Max}\left\{H_1^{(2)}(\underline{x}), H_2^{(2)}(\underline{x})\right\} + 12.65$$

FINAL UNWEIGHTED P-FUNCTIONS (STANDARDIZATION REMOVED) ********** TABLE 18 **********

P-FUNCTION NO. 1

IGR	IRCI	IPR	IOH	CONST
2.811E-01	8.056E-01	7.073E-01	8.906E-02	-8.481E+00
1.472E+00	5.202E-01	3.936E-02	4.275E-02	-8.975E+00

WEIGHT = -2.480E+00

P-FUNCTION NO. 2

IGR	IRCI	IPR	IOH	CONST
2.868E-01	-7.288E-01	5.417E-01	-9.429E-03	1.256E+00
-2.098E-01	8.604E-01	-6.547E-01	1.037E-02	-2.342E+00

WEIGHT = -1.512E+00

CONSTANT TERM = 1.265E+01

P-FUNCTIONS SAVED ON UNIT 30 ==========

FINAL PREDICTED VALUES AND ERRORS

IGR	IRCI	IPR	IOH	VEL	PRED	ERROR
2.000	4.000	1.000	1.000	23.39	22.18	1.206
2.000	4.000	1.000	2.000	23.38	22.03	1.352
2.000	4.000	1.000	3.000	22.08	21.79	.2883
2.000	4.000	1.000	4.000	21.18	21.56	-.3751
2.000	4.000	2.000	1.000	23.39	20.58	2.808
2.000	4.000	2.000	2.000	23.38	20.38	3.005
2.000	4.000	2.000	3.000	22.08	20.17	1.911
2.000	4.000	2.000	4.000	21.18	19.96	1.218
2.000	4.000	3.000	1.000	19.86	18.01	1.851
2.000	4.000	3.000	2.000	19.85	17.80	2.048
2.000	4.000	3.000	3.000	19.01	17.60	1.414
2.000	4.000	3.000	4.000	18.34	17.39	.9511
2.000	4.000	4.000	1.000	14.10	15.44	-1.335
2.000	4.000	4.000	2.000	14.10	15.23	-1.129
2.000	4.000	4.000	3.000	13.79	15.02	-1.232
2.000	4.000	4.000	4.000	13.44	14.82	-1.376
2.000	4.000	5.000	1.000	11.00	12.86	-1.862
2.000	4.000	5.000	2.000	11.00	12.66	-1.656
2.000	4.000	5.000	3.000	10.86	12.45	-1.589
2.000	4.000	5.000	4.000	10.65	12.24	-1.592
3.000	4.000	1.000	1.000	19.03	18.85	.1797
3.000	4.000	1.000	2.000	19.01	18.73	.2814
3.000	4.000	1.000	3.000	18.22	18.61	-.3868
3.000	4.000	1.000	4.000	17.60	18.49	-.8851
3.000	4.000	2.000	1.000	19.03	18.07	.9564
3.000	4.000	2.000	2.000	19.01	17.98	1.028
3.000	4.000	2.000	3.000	18.22	17.89	.3299
3.000	4.000	2.000	4.000	17.60	17.80	-.1983
3.000	4.000	3.000	1.000	16.62	16.88	-.2580
3.000	4.000	3.000	2.000	16.61	16.67	-6.1394E-02
3.000	4.000	3.000	3.000	16.08	16.46	-.3848
3.000	4.000	3.000	4.000	15.60	16.26	-.6581
3.000	4.000	4.000	1.000	13.18	14.30	-1.125
3.000	4.000	4.000	2.000	13.17	14.10	-.9281
3.000	4.000	4.000	3.000	12.91	13.89	-.9814
3.000	4.000	4.000	4.000	12.60	13.68	-1.085
3.000	4.000	5.000	1.000	11.00	11.73	-.7314
3.000	4.000	5.000	2.000	11.00	11.52	-.5248
3.000	4.000	5.000	3.000	10.86	11.32	-.4581
3.000	4.000	5.000	4.000	10.65	11.11	-.4615
4.000	4.000	1.000	1.000	15.45	14.91	.5445
4.000	4.000	1.000	2.000	15.44	14.81	.6262
4.000	4.000	1.000	3.000	14.97	14.72	.2480
4.000	4.000	1.000	4.000	14.56	14.63	-7.0204E-02
4.000	4.000	2.000	1.000	15.45	13.99	1.461
4.000	4.000	2.000	2.000	15.44	13.90	1.543
4.000	4.000	2.000	3.000	14.97	13.81	1.165
4.000	4.000	2.000	4.000	14.56	13.71	.8463
4.000	4.000	3.000	1.000	13.82	13.07	.7475
4.000	4.000	3.000	2.000	13.82	12.98	.8393
4.000	4.000	3.000	3.000	13.50	12.89	.6111
4.000	4.000	3.000	4.000	13.16	12.80	.3628
4.000	4.000	4.000	1.000	11.35	12.16	-.8060
4.000	4.000	4.000	2.000	11.35	12.06	-.7142
4.000	4.000	4.000	3.000	11.19	11.97	-.7824

(VEL-PRED in this case)

4,000	4,000	4,000	4,000	10,96	11,88	-,9206
4,000	4,000	5,000	1,000	9,860	10,60	-,7406
4,000	4,000	5,000	2,000	9,860	10,39	-,5340
4,000	4,000	5,000	3,000	9,770	10,19	-,4174
4,000	4,000	5,000	4,000	9,590	9,981	-,3907
5,000	4,000	1,000	1,000	10,19	10,82	-,6309
5,000	4,000	1,000	2,000	10,19	10,73	-,5391
5,000	4,000	1,000	3,000	10,000	10,64	-,6373
5,000	4,000	1,000	4,000	9,850	10,55	-,6956
5,000	4,000	2,000	1,000	10,19	9,904	,2856
5,000	4,000	2,000	2,000	10,19	9,813	,3774
5,000	4,000	2,000	3,000	10,000	9,721	,2792
5,000	4,000	2,000	4,000	9,850	9,629	,2210
5,000	4,000	3,000	1,000	9,460	8,988	,4722
5,000	4,000	3,000	2,000	9,450	8,896	,5539
5,000	4,000	3,000	3,000	9,320	8,804	,5157
5,000	4,000	3,000	4,000	9,190	8,713	,4775
5,000	4,000	4,000	1,000	8,230	8,071	,1587
5,000	4,000	4,000	2,000	8,230	7,980	,2505
5,000	4,000	4,000	3,000	8,160	7,888	,2722
5,000	4,000	4,000	4,000	8,060	7,796	,2640
5,000	4,000	5,000	1,000	7,420	7,155	,2652
5,000	4,000	5,000	2,000	7,420	7,063	,3570
5,000	4,000	5,000	3,000	7,370	6,971	,3987
5,000	4,000	5,000	4,000	7,300	6,879	,4205
6,000	4,000	1,000	1,000	4,880	6,736	-1,856
6,000	4,000	1,000	2,000	4,870	6,644	-1,774
6,000	4,000	1,000	3,000	4,820	6,553	-1,733
6,000	4,000	1,000	4,000	4,790	6,461	-1,671
6,000	4,000	2,000	1,000	4,880	5,820	-,9397
6,000	4,000	2,000	2,000	4,870	5,728	-,8580
6,000	4,000	2,000	3,000	4,820	5,636	-,8162
6,000	4,000	2,000	4,000	4,790	5,544	-,7544
6,000	4,000	3,000	1,000	4,700	4,903	-,2032
6,000	4,000	3,000	2,000	4,700	4,811	-,1114
6,000	4,000	3,000	3,000	4,660	4,720	-5,9660E-02
6,000	4,000	3,000	4,000	4,630	4,628	2,1142E-03
6,000	4,000	4,000	1,000	4,380	3,987	,3933
6,000	4,000	4,000	2,000	4,380	3,895	,4851
6,000	4,000	4,000	3,000	4,350	3,803	,5469
6,000	4,000	4,000	4,000	4,330	3,711	,6186
6,000	4,000	5,000	1,000	4,140	3,070	1,070
6,000	4,000	5,000	2,000	4,130	2,978	1,152
6,000	4,000	5,000	3,000	4,120	2,887	1,233
6,000	4,000	5,000	4,000	4,100	2,795	1,305
2,000	5,000	1,000	1,000	20,03	18,97	1,064
2,000	5,000	1,000	2,000	19,97	18,73	1,241
2,000	5,000	1,000	3,000	19,08	18,49	,5872
2,000	5,000	1,000	4,000	18,38	18,26	,1237
2,000	5,000	2,000	1,000	20,03	18,20	1,829
2,000	5,000	2,000	2,000	19,97	17,96	2,005
2,000	5,000	2,000	3,000	19,08	17,73	1,352
2,000	5,000	2,000	4,000	18,38	17,49	,8884
2,000	5,000	3,000	1,000	18,67	17,11	1,558
2,000	5,000	3,000	2,000	18,66	16,91	1,754
2,000	5,000	3,000	3,000	17,95	16,70	1,251
2,000	5,000	3,000	4,000	17,35	16,49	,8575
2,000	5,000	4,000	1,000	14,10	14,54	-,4391
2,000	5,000	4,000	2,000	14,10	14,33	-,2324
2,000	5,000	4,000	3,000	13,79	14,15	-,3358
2,000	5,000	4,000	4,000	13,44	13,92	-,4792
2,000	5,000	5,000	1,000	11,00	11,97	-,9658

2,000	5,000	5,000	2,000	11,00	11,76	-,7591
2,000	5,000	5,000	3,000	10,86	11,55	-,6925
2,000	5,000	5,000	4,000	10,65	11,35	-,6959
3,000	5,000	1,000	1,000	16,56	16,26	,3006
3,000	5,000	1,000	2,000	16,53	16,14	,3923
3,000	5,000	1,000	3,000	15,98	16,02	-3,5946E-02
3,000	5,000	1,000	4,000	15,50	15,89	-,3942
3,000	5,000	2,000	1,000	16,56	17,15	-,5914
3,000	5,000	2,000	2,000	16,53	17,03	-,4997
3,000	5,000	2,000	3,000	15,98	16,91	-,9280
3,000	5,000	2,000	4,000	15,50	16,79	-1,286
3,000	5,000	3,000	1,000	15,64	15,98	-,3416
3,000	5,000	3,000	2,000	15,64	15,77	-,1350
3,000	5,000	3,000	3,000	15,19	15,57	-,3783
3,000	5,000	3,000	4,000	14,77	15,36	-,5917
3,000	5,000	4,000	1,000	12,55	13,41	-,8583
3,000	5,000	4,000	2,000	12,55	13,20	-,6517
3,000	5,000	4,000	3,000	12,33	13,00	-,6650
3,000	5,000	4,000	4,000	12,05	12,79	-,7384
3,000	5,000	5,000	1,000	10,75	10,83	-8,4989E-02
3,000	5,000	5,000	2,000	10,75	10,63	,1216
3,000	5,000	5,000	3,000	10,62	10,42	,1983
3,000	5,000	5,000	4,000	10,42	10,22	,2049
4,000	5,000	1,000	1,000	13,93	12,93	1,005
4,000	5,000	1,000	2,000	13,91	12,80	1,106
4,000	5,000	1,000	3,000	13,56	12,68	,8780
4,000	5,000	1,000	4,000	13,21	12,56	,6497
4,000	5,000	2,000	1,000	13,93	13,80	,1295
4,000	5,000	2,000	2,000	13,91	13,70	,2142
4,000	5,000	2,000	3,000	13,56	13,57	-1,4085E-02
4,000	5,000	2,000	4,000	13,21	13,45	-,2424
4,000	5,000	3,000	1,000	13,31	12,88	,4260
4,000	5,000	3,000	2,000	13,30	12,79	,5078
4,000	5,000	3,000	3,000	13,01	12,70	,3095
4,000	5,000	3,000	4,000	12,70	12,61	9,1509E-02
4,000	5,000	4,000	1,000	11,00	11,97	-,9675
4,000	5,000	4,000	2,000	11,00	11,88	-,8757
4,000	5,000	4,000	3,000	10,85	11,78	-,9339
4,000	5,000	4,000	4,000	10,64	11,66	-1,018
4,000	5,000	5,000	1,000	9,600	9,704	-,1042
4,000	5,000	5,000	2,000	9,590	9,498	9,2426E-02
4,000	5,000	5,000	3,000	9,510	9,291	,2191
4,000	5,000	5,000	4,000	9,340	9,084	,2552
5,000	5,000	1,000	1,000	9,320	9,591	-,2715
5,000	5,000	1,000	2,000	9,310	9,470	-,1598
5,000	5,000	1,000	3,000	9,160	9,348	-,1881
5,000	5,000	1,000	4,000	9,040	9,226	-,1864
5,000	5,000	2,000	1,000	9,320	9,716	-,3959
5,000	5,000	2,000	2,000	9,310	9,624	-,3141
5,000	5,000	2,000	3,000	9,160	9,532	-,3723
5,000	5,000	2,000	4,000	9,040	9,441	-,4006
5,000	5,000	3,000	1,000	9,070	8,799	,2706
5,000	5,000	3,000	2,000	9,070	8,708	,3624
5,000	5,000	3,000	3,000	8,940	8,616	,3242
5,000	5,000	3,000	4,000	8,830	8,524	,3059
5,000	5,000	4,000	1,000	7,940	7,883	5,7146E-02
5,000	5,000	4,000	2,000	7,940	7,791	,1489
5,000	5,000	4,000	3,000	7,870	7,699	,1707
5,000	5,000	4,000	4,000	7,780	7,608	,1725
5,000	5,000	5,000	1,000	7,180	6,966	,2137
5,000	5,000	5,000	2,000	7,180	6,875	,3054
5,000	5,000	5,000	3,000	7,140	6,783	,3572

5,000	5,000	5,000	4,000	7,070	6,691	,3790
6,000	5,000	1,000	1,000	4,430	6,258	-1,828
6,000	5,000	1,000	2,000	4,420	6,136	-1,716
6,000	5,000	1,000	3,000	4,390	6,014	-1,624
6,000	5,000	1,000	4,000	4,360	5,892	-1,532
6,000	5,000	2,000	1,000	4,430	5,631	-1,201
6,000	5,000	2,000	2,000	4,420	5,539	-1,119
6,000	5,000	2,000	3,000	4,390	5,448	-1,058
6,000	5,000	2,000	4,000	4,360	5,356	-,9959
6,000	5,000	3,000	1,000	4,390	4,715	-,3247
6,000	5,000	3,000	2,000	4,390	4,623	-,2330
6,000	5,000	3,000	3,000	4,360	4,531	-,1712
6,000	5,000	3,000	4,000	4,330	4,439	-,1094
6,000	5,000	4,000	1,000	4,110	3,798	,3118
6,000	5,000	4,000	2,000	4,100	3,706	,3936
6,000	5,000	4,000	3,000	4,080	3,615	,4653
6,000	5,000	4,000	4,000	4,060	3,523	,5371
6,000	5,000	5,000	1,000	3,890	2,882	1,008
6,000	5,000	5,000	2,000	3,890	2,790	1,100
6,000	5,000	5,000	3,000	3,880	2,698	1,182
6,000	5,000	5,000	4,000	3,860	2,606	1,254
2,000	6,000	1,000	1,000	14,51	15,67	-1,157
2,000	6,000	1,000	2,000	14,49	15,43	-,9406
2,000	6,000	1,000	3,000	14,12	15,19	-1,074
2,000	6,000	1,000	4,000	13,74	14,96	-1,217
2,000	6,000	2,000	1,000	14,51	14,90	-,3924
2,000	6,000	2,000	2,000	14,49	14,67	-,1758
2,000	6,000	2,000	3,000	14,12	14,43	-,3093
2,000	6,000	2,000	4,000	13,74	14,19	-,4527
2,000	6,000	3,000	1,000	14,51	14,14	,3723
2,000	6,000	3,000	2,000	14,49	13,90	,5889
2,000	6,000	3,000	3,000	14,12	13,66	,4554
2,000	6,000	3,000	4,000	13,74	13,43	,3120
2,000	6,000	4,000	1,000	12,98	13,37	-,3930
2,000	6,000	4,000	2,000	12,97	13,14	-,1664
2,000	6,000	4,000	3,000	12,72	12,90	-,1798
2,000	6,000	4,000	4,000	12,42	12,66	-,2433
2,000	6,000	5,000	1,000	11,00	11,07	-6,9354E-02
2,000	6,000	5,000	2,000	11,00	10,86	,1373
2,000	6,000	5,000	3,000	10,86	10,66	,2039
2,000	6,000	5,000	4,000	10,65	10,45	,2005
3,000	6,000	1,000	1,000	13,70	13,67	3,1544E-02
3,000	6,000	1,000	2,000	13,69	13,55	,1432
3,000	6,000	1,000	3,000	13,38	13,43	-4,5044E-02
3,000	6,000	1,000	4,000	13,05	13,30	-,2533
3,000	6,000	2,000	1,000	13,70	14,52	-,8223
3,000	6,000	2,000	2,000	13,69	14,29	-,5958
3,000	6,000	2,000	3,000	13,38	14,05	-,6692
3,000	6,000	2,000	4,000	13,05	13,81	-,7626
3,000	6,000	3,000	1,000	13,70	13,76	-5,7614E-02
3,000	6,000	3,000	2,000	13,69	13,52	,1690
3,000	6,000	3,000	3,000	13,38	13,28	9,5519E-02
3,000	6,000	3,000	4,000	13,05	13,05	2,0853E-03
3,000	6,000	4,000	1,000	12,34	12,51	-,1719
3,000	6,000	4,000	2,000	12,34	12,31	3,4748E-02
3,000	6,000	4,000	3,000	12,13	12,10	3,1383E-02
3,000	6,000	4,000	4,000	11,86	11,89	-3,1983E-02
3,000	6,000	5,000	1,000	10,60	9,939	,6614
3,000	6,000	5,000	2,000	10,59	9,732	,8581
3,000	6,000	5,000	3,000	10,47	9,525	,9447
3,000	6,000	5,000	4,000	10,27	9,319	,9513
4,000	6,000	1,000	1,000	11,27	10,33	,9355

4,000	6,000	1,000	2,000	11,26	10,21	1,047
4,000	6,000	1,000	3,000	11,08	10,09	,9889
4,000	6,000	1,000	4,000	10,85	9,969	,8806
4,000	6,000	2,000	1,000	11,27	11,23	4,3405E-02
4,000	6,000	2,000	2,000	11,26	11,10	,1551
4,000	6,000	2,000	3,000	11,08	10,98	9,6817E-02
4,000	6,000	2,000	4,000	10,85	10,86	-1,1477E-02
4,000	6,000	3,000	1,000	11,27	12,12	-,8487
4,000	6,000	3,000	2,000	11,26	12,00	-,7370
4,000	6,000	3,000	3,000	11,08	11,88	-,7953
4,000	6,000	3,000	4,000	10,85	11,75	-,9035
4,000	6,000	4,000	1,000	10,35	11,38	-1,031
4,000	6,000	4,000	2,000	10,34	11,17	-,8345
4,000	6,000	4,000	3,000	10,23	10,97	-,7378
4,000	6,000	4,000	4,000	10,03	10,76	-,7312
4,000	6,000	5,000	1,000	9,090	8,808	,2822
4,000	6,000	5,000	2,000	9,090	8,601	,4888
4,000	6,000	5,000	3,000	9,020	8,395	,6255
4,000	6,000	5,000	4,000	8,880	8,188	,6921
5,000	6,000	1,000	1,000	7,790	7,001	,7894
5,000	6,000	1,000	2,000	7,780	6,879	,9011
5,000	6,000	1,000	3,000	7,700	6,757	,9428
5,000	6,000	1,000	4,000	7,620	6,635	,9845
5,000	6,000	2,000	1,000	7,790	7,893	-,1027
5,000	6,000	2,000	2,000	7,780	7,771	9,0417E-03
5,000	6,000	2,000	3,000	7,700	7,649	5,0748E-02
5,000	6,000	2,000	4,000	7,620	7,528	9,2454E-02
5,000	6,000	3,000	1,000	7,790	8,611	-,8209
5,000	6,000	3,000	2,000	7,780	8,519	-,7391
5,000	6,000	3,000	3,000	7,700	8,427	-,7274
5,000	6,000	3,000	4,000	7,620	8,336	-,7156
5,000	6,000	4,000	1,000	7,370	7,694	-,3244
5,000	6,000	4,000	2,000	7,370	7,603	-,2326
5,000	6,000	4,000	3,000	7,310	7,511	-,2008
5,000	6,000	4,000	4,000	7,240	7,419	-,1791
5,000	6,000	5,000	1,000	6,710	6,778	-6,7862E-02
5,000	6,000	5,000	2,000	6,710	6,686	2,3912E-02
5,000	6,000	5,000	3,000	6,670	6,594	7,5686E-02
5,000	6,000	5,000	4,000	6,620	6,503	,1175
6,000	6,000	1,000	1,000	3,620	3,667	-4,6664E-02
6,000	6,000	1,000	2,000	3,620	3,545	7,5042E-02
6,000	6,000	1,000	3,000	3,600	3,423	,1767
6,000	6,000	1,000	4,000	3,580	3,302	,2785
6,000	6,000	2,000	1,000	3,620	4,559	-,9387
6,000	6,000	2,000	2,000	3,620	4,437	-,8170
6,000	6,000	2,000	3,000	3,600	4,315	-,7153
6,000	6,000	2,000	4,000	3,580	4,194	-,6136
6,000	6,000	3,000	1,000	3,620	4,526	-,9063
6,000	6,000	3,000	2,000	3,620	4,434	-,8145
6,000	6,000	3,000	3,000	3,600	4,343	-,7427
6,000	6,000	3,000	4,000	3,580	4,251	-,6709
6,000	6,000	4,000	1,000	3,550	3,610	-5,9747E-02
6,000	6,000	4,000	2,000	3,550	3,518	3,2027E-02
6,000	6,000	4,000	3,000	3,530	3,426	,1038
6,000	6,000	4,000	4,000	3,510	3,334	,1756
6,000	6,000	5,000	1,000	3,390	2,693	,6968
6,000	6,000	5,000	2,000	3,390	2,601	,7885
6,000	6,000	5,000	3,000	3,370	2,510	,8603
6,000	6,000	5,000	4,000	3,360	2,418	,9421
2,000	7,000	1,000	1,000	10,71	12,37	-1,658
2,000	7,000	1,000	2,000	10,69	12,13	-1,442
2,000	7,000	1,000	3,000	10,56	11,90	-1,335

2,000	7,000	1,000	4,000	10,35	11,66	-1,309
2,000	7,000	2,000	1,000	10,71	11,60	-,8936
2,000	7,000	2,000	2,000	10,69	11,37	-,6770
2,000	7,000	2,000	3,000	10,56	11,13	-,5704
2,000	7,000	2,000	4,000	10,35	10,89	-,5439
2,000	7,000	3,000	1,000	10,71	10,84	-,1288
2,000	7,000	3,000	2,000	10,69	10,60	8,7726E-02
2,000	7,000	3,000	3,000	10,56	10,37	,1943
2,000	7,000	3,000	4,000	10,35	10,13	,2209
2,000	7,000	4,000	1,000	10,71	10,07	,6359
2,000	7,000	4,000	2,000	10,69	9,838	,8524
2,000	7,000	4,000	3,000	10,56	9,601	,9590
2,000	7,000	4,000	4,000	10,35	9,364	,9856
2,000	7,000	5,000	1,000	10,46	9,309	1,151
2,000	7,000	5,000	2,000	10,46	9,073	1,387
2,000	7,000	5,000	3,000	10,33	8,836	1,494
2,000	7,000	5,000	4,000	10,14	8,600	1,540
3,000	7,000	1,000	1,000	9,790	11,08	-1,288
3,000	7,000	1,000	2,000	9,780	10,96	-1,176
3,000	7,000	1,000	3,000	9,680	10,83	-1,154
3,000	7,000	1,000	4,000	9,500	10,71	-1,212
3,000	7,000	2,000	1,000	9,790	11,22	-1,433
3,000	7,000	2,000	2,000	9,780	10,99	-1,207
3,000	7,000	2,000	3,000	9,680	10,75	-1,070
3,000	7,000	2,000	4,000	9,500	10,51	-1,014
3,000	7,000	3,000	1,000	9,790	10,46	-,6688
3,000	7,000	3,000	2,000	9,780	10,22	-,4422
3,000	7,000	3,000	3,000	9,680	9,986	-,3056
3,000	7,000	3,000	4,000	9,500	9,749	-,2491
3,000	7,000	4,000	1,000	9,790	9,694	9,5954E-02
3,000	7,000	4,000	2,000	9,780	9,457	,3225
3,000	7,000	4,000	3,000	9,680	9,221	,4591
3,000	7,000	4,000	4,000	9,500	8,984	,5157
3,000	7,000	5,000	1,000	9,600	8,929	,6707
3,000	7,000	5,000	2,000	9,590	8,693	,8972
3,000	7,000	5,000	3,000	9,500	8,456	1,044
3,000	7,000	5,000	4,000	9,330	8,220	1,110
4,000	7,000	1,000	1,000	8,480	7,744	,7364
4,000	7,000	1,000	2,000	8,470	7,622	,8481
4,000	7,000	1,000	3,000	8,410	7,500	,9098
4,000	7,000	1,000	4,000	8,280	7,379	,9015
4,000	7,000	2,000	1,000	8,480	8,636	-,1557
4,000	7,000	2,000	2,000	8,470	8,514	-4,3987E-02
4,000	7,000	2,000	3,000	8,410	8,392	1,7719E-02
4,000	7,000	2,000	4,000	8,280	8,271	9,4247E-03
4,000	7,000	3,000	1,000	8,480	9,528	-1,048
4,000	7,000	3,000	2,000	8,470	9,406	-,9361
4,000	7,000	3,000	3,000	8,410	9,284	-,8744
4,000	7,000	3,000	4,000	8,280	9,163	-,8826
4,000	7,000	4,000	1,000	8,480	9,314	-,8340
4,000	7,000	4,000	2,000	8,470	9,077	-,6074
4,000	7,000	4,000	3,000	8,410	8,841	-,4308
4,000	7,000	4,000	4,000	8,280	8,604	-,3243
4,000	7,000	5,000	1,000	8,360	7,911	,4486
4,000	7,000	5,000	2,000	8,350	7,705	,6453
4,000	7,000	5,000	3,000	8,300	7,498	,8019
4,000	7,000	5,000	4,000	8,180	7,291	,8885
5,000	7,000	1,000	1,000	6,020	4,410	1,610
5,000	7,000	1,000	2,000	6,020	4,288	1,732
5,000	7,000	1,000	3,000	5,990	4,166	1,824
5,000	7,000	1,000	4,000	5,940	4,045	1,895
5,000	7,000	2,000	1,000	6,020	5,302	,7182

5.000	7.000	2.000	2.000	6.020	5.180	.8399
5.000	7.000	2.000	3.000	5.990	5.058	.9316
5.000	7.000	2.000	4.000	5.940	4.937	1.003
5.000	7.000	3.000	1.000	6.020	6.194	-.1738
5.000	7.000	3.000	2.000	6.020	6.072	-5.2126E-02
5.000	7.000	3.000	3.000	5.990	5.950	3.9580E-02
5.000	7.000	3.000	4.000	5.940	5.829	.1113
5.000	7.000	4.000	1.000	6.020	7.086	-1.066
5.000	7.000	4.000	2.000	6.020	6.964	-.9442
5.000	7.000	4.000	3.000	5.990	6.842	-.8525
5.000	7.000	4.000	4.000	5.940	6.721	-.7808
5.000	7.000	5.000	1.000	6.010	6.589	-.5794
5.000	7.000	5.000	2.000	6.010	6.498	-.4876
5.000	7.000	5.000	3.000	5.980	6.367	-.3873
5.000	7.000	5.000	4.000	5.930	6.161	-.2307
6.000	7.000	1.000	1.000	2.440	1.076	1.364
6.000	7.000	1.000	2.000	2.430	.9541	1.476
6.000	7.000	1.000	3.000	2.430	.8323	1.598
6.000	7.000	1.000	4.000	2.420	.7106	1.709
6.000	7.000	2.000	1.000	2.440	1.968	.4722
6.000	7.000	2.000	2.000	2.430	1.846	.5839
6.000	7.000	2.000	3.000	2.430	1.724	.7056
6.000	7.000	2.000	4.000	2.420	1.603	.8173
6.000	7.000	3.000	1.000	2.440	2.860	-.4199
6.000	7.000	3.000	2.000	2.430	2.738	-.3082
6.000	7.000	3.000	3.000	2.430	2.616	-.1865
6.000	7.000	3.000	4.000	2.420	2.495	-7.4783E-02
6.000	7.000	4.000	1.000	2.440	3.421	-.9813
6.000	7.000	4.000	2.000	2.430	3.330	-.8995
6.000	7.000	4.000	3.000	2.430	3.238	-.8077
6.000	7.000	4.000	4.000	2.420	3.146	-.7260
6.000	7.000	5.000	1.000	2.440	2.505	-6.4756E-02
6.000	7.000	5.000	2.000	2.430	2.413	1.7018E-02
6.000	7.000	5.000	3.000	2.430	2.321	.1088
6.000	7.000	5.000	4.000	2.420	2.229	.1906

OVERALL STATISTICS ********** TABLE 2 **********

NDIM = 4 NSAMP = 400

	IGR	IRCI	IPR	IOH	VEL	PREDIC
MEANS =	4.00	5.50	3.00	2.50	9.97	9.97
SD =	1.41	1.12	1.41	1.12	4.64	4.57
COVARIANCE=	2.00	5.568E-13	0.	4.657E-14	-5.36	-5.33
	5.568E-13	1.25	-6.519E-13	7.451E-13	-1.90	-1.90
	0.	-6.519E-13	2.00	-1.863E-13	-1.56	-1.57
	4.657E-14	7.451E-13	-1.863E-13	1.25	-.188	-.189
	-5.36	-1.90	-1.56	-.188	21.6	20.9
	-5.33	-1.90	-1.57	-.189	20.9	20.9
CORR =	1.000	.000	0.000	.000	-.816	-.825
	.000	1.000	-.000	.000	-.367	-.373
	0.000	-.000	1.000	-.000	-.238	-.244
	.000	.000	-.000	1.000	-.036	-.037
	-.816	-.367	-.238	-.036	1.000	.984
	-.825	-.373	-.244	-.037	.984	1.000

NET HYPERPLANE STATISTICS FOR 4 REGIONS ********** TABLE 3 **********

REGION NO.	I.D.	POPULATION	POP (PCT.)	RMS	MAXERROR	PCTVE
1	2 1	134	33.5	.7047	1.86	97.0
2	2 2	104	26.0	.8564	1.90	96.6
3	1 1	87	21.7	.9704	3.00	93.1
4	1 2	75	18.7	.8700	2.01	94.2

Note 1.

Note 1. This display indicates which hyperplanes are involved in the various regions. Each column pertains to one of the P-functions. The left-most column applies to P-function #1; the next column to P-function #2; and so on. There are as many columns as there are P-functions in the fit. (Up to 9 P-functions are allowed.)

Within each column, the numbers identify which hyperplane of that P-function is involved. Numeral "1" means hyperplane #1, etc. (Up to 10 hyperplanes per P-function are allowed.)

As an example, in the above output, Region 4 involves hyperplane #1 of the 1st P-function and hyperplane #2 of the 2nd P-function.

QUALITATIVE CHARACTERIZATION OF SUBREGIONS ********** TABLE 4 **********

	IGR	IRCI	IPR	IOH	VEL	PREDIC
REGION 1	2	-2	1	-1	-2	-2
REGION 2	1	2	-2	1	-1	-1
REGION 3	-2	-1	2	1	2	2
REGION 4	-3	2	-1	1	2	2

EXPLICIT CHARACTERIZATION OF SUBREGIONS ********** TABLE 5 **********

RG	PCT.		IGR	IRCI	IPR	IOH	VEL	PREDIC
1	33.5	MEAN	5.2	4.9	3.3	2.5	7.6	7.6
		S.D.	.83	.95	1.3	1.1	4.0	3.9
		MAX	6.0	7.0	5.0	4.0	19.	18.
		MIN	3.0	4.0	1.0	1.0	2.4	2.2
2	26.0	MEAN	4.5	6.1	1.7	2.5	8.8	8.8
		S.D.	1.1	.91	.86	1.1	4.7	4.8
		MAX	6.0	7.0	4.0	4.0	23.	22.
		MIN	2.0	4.0	1.0	1.0	2.4	.71
3	21.7	MEAN	2.9	5.0	4.2	2.5	13.	13.
		S.D.	.83	.93	.90	1.1	3.7	3.3
		MAX	5.0	7.0	5.0	4.0	23.	21.
		MIN	2.0	4.0	2.0	1.0	5.9	6.2
4	18.7	MEAN	2.4	6.3	2.8	2.5	13.	13.
		S.D.	.59	.82	1.3	1.1	3.6	3.4
		MAX	4.0	7.0	5.0	4.0	23.	22.
		MIN	2.0	4.0	1.0	1.0	8.3	8.2

PARAMETERS OF FIT FOR ACTUAL (NON-STANDARDIZED) DATA ********** TABLE 6 **********

REG	PCT.	IGR	IRCI	IPR	IOH	CONSTANT
1	33.5	-4.1	-.19	-.92	-9.18E-02	33.
2	26.0	-3.3	-2.6	.89	-.12	38.
3	21.7	-1.1	-.90	-2.6	-.21	32.
4	18.7	-.38	-3.3	-.76	-.24	37.

Note:

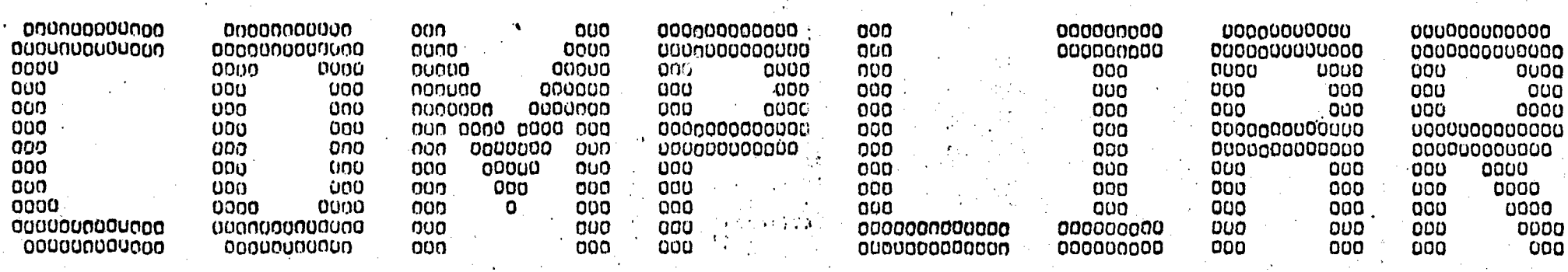
In region 1, the equation for the dependent variable is:

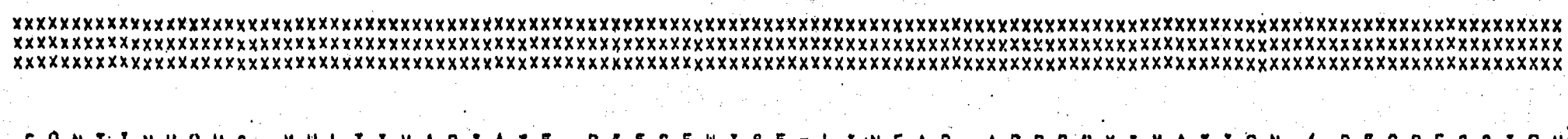
$$\text{Mobil} = -4.1 * \text{IGR} - 0.19 * \text{IRCI} - 0.92 * \text{IPR} - (9.18 \times 10^{-2}) * \text{IOH} + 33$$

The equations relating the variables in the other regions are similarly obtained from the above table.

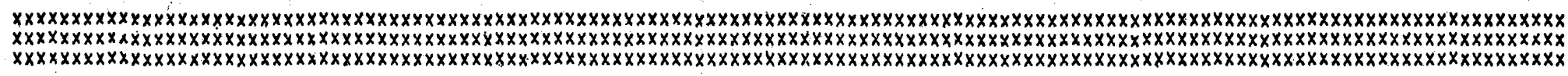
PARAMETERS OF FIT FOR STANDARDIZED INDEPENDENT VARIABLES ********** TABLE 7 **********

REG	PCT.	IGR	IRCI	IPR	IOH	CONSTANT
1	33,5	-5,777	-,211	-1,296	-,103	12,652
2	26,0	-4,715	-2,897	1,262	-,136	13,232
3	21,7	-1,599	-1,002	-3,639	-,231	14,093
4	18,7	-,538	-3,688	-1,081	-,264	14,672



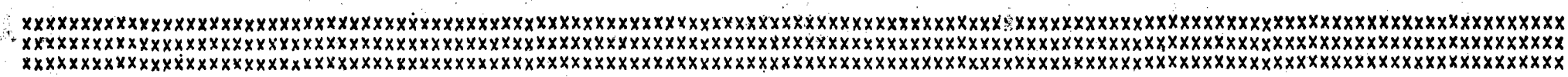


CONTINUOUS MULTIVARIATE PIECEWISE-LINEAR APPROXIMATION / REGRESSION



DEVELOPED BY: TECHNOLOGY SERVICE CORPORATION
2811 WILSHIRE BOULEVARD
SANTA MONICA, CALIFORNIA 90403

FOR: U.S. ARMY TANK AUTOMOTIVE RESEARCH
AND DEVELOPMENT COMMAND (TARADCOM)
WARREN, MICHIGAN



5.2 AN 8-DIMENSIONAL EXAMPLE

The COMPLIAR software was used to furnish a repro-model for an 8-dimensional case. The example utilized 5184 data points consisting of values for the 8 input variables (4 terrain parameters and 4 vehicle-related parameters) together with the associated values of tank velocity. These data were supplied using the '71 Mobility Model.

A repro-model consisting of 2 P-functions with 2 hyperplanes per P-functions was capable of accounting for more than 94% of the variation between the input/output data.

The final repro-model is summarized in Table 5.2 below: Note the simplicity of the result. The repro-model given in Table 5.2 can easily be used to furnish input/output data on a hand calculator.

Table 5.2 Repro-Model for 8-Dimensional Case

FINAL WEIGHTED P-FUNCTIONS (STANDARDIZATION REMOVED, MAGNITUDE OF P-FUNCTION WEIGHTS ABSORBED) ********** TABLE 1A **********

FINAL ERROR = 1.31 (BIASED) 1.31 (UNBIASED)

FINAL PCTVE = 94.5 (BIASED) 94.5 (UNBIASED)

P-FUNCTION NO. 1 SIGN OF WEIGHT IS -

	x_1 RMS	x_2 GRADE	x_3 TR FRC	x_4 GVW	x_5 RCI	x_6 VRIDE	x_7 OH	x_8 VOOB	HYPCONST
$H_1^{(1)}(\underline{x})$ =	3.177E-01	3.986E-01	-5.954E-04	1.030E-04	-3.799E-02	-1.193E-02	6.854E-02	-2.395E-02	-3.774E+00
$H_2^{(1)}(\underline{x})$ =	3.708E+00	1.950E-01	-1.401E-04	2.543E-05	-6.988E-03	-3.574E-01	6.695E-02	-2.666E-02	-3.233E+00

P-FUNCTION NO. 2 SIGN OF WEIGHT IS +

$H_1^{(2)}(\underline{x})$ =	-3.815E-01	-1.178E-01	6.487E-05	-9.474E-06	1.252E-02	4.186E-02	-5.862E-02	1.633E-02	3.825E-01
$H_2^{(2)}(\underline{x})$ =	6.023E-01	7.651E-02	-2.483E-05	3.665E-06	-7.984E-03	-5.206E-02	4.208E-02	-1.211E-02	-7.393E-01

CONSTANT TERM = 1.448E+01

As an example, the equation for $H_2^{(1)}(\underline{x})$ is:

$$H_2^{(1)}(\underline{x}) = 3.708x_1 + (1.95\times10^{-1})x_2 - (1.401\times10^{-4})x_3 + (2.543\times10^{-5})x_4 - (6.988\times10^{-3})x_5$$
$$- (3.574\times10^{-1})x_6 + (6.695\times10^{-2})x_7 - (2.666\times10^{-2})x_8 - 3.233.$$

Since the magnitude of the P-function weights have now been absorbed, the sign of the P-function weights are now employed in writing the final form for the dependent variable:

$$VEL(\underline{x}) = (-1)\cdot\max\left\{H_1^{(1)}(\underline{x}),\ H_2^{(1)}(\underline{x})\right\} + (+1)\cdot\max\left\{H_1^{(2)}(\underline{x}),\ H_2^{(2)}(\underline{x})\right\} + 14.48$$

where $\underline{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8]$

VI. PROGRAM DESCRIPTION

The COMPLIAR program has a structure composed of a few key elements and several supporting elements. This structure becomes evident when a top-down form of documentation is used. Top-down documentation consists of describing the major functions of the program and indenting at various levels to indicate the supporting structure. This type of documentation is analogous to the kind of outline used in organizing a text, where major ideas and their supporting discussions are clearly evidenced.

The program was written in IFTRAN, which is a TSC superset of FORTRAN. IFTRAN is a structured programming language and has the facility to arrange comment cards in the same structure as the IFTRAN program listing. When appropriate comment cards are inserted, this produces what we call a "Commented IFTRAN Listing." For some subroutines, the commented IFTRAN listing is an excellent way to functionally describe the program structure. Where appropriate, this has been used.

Section 6.1 describes the main program. Section 6.2 describes the routines used for inputting data and other required inputs. Section 6.3 describes the optimization routines. The statistical analysis routines are described in Section 6.4. These routines perform the analysis by regions.

As mentioned earlier, complete annotated listings of the program can be obtained from the Science and Technology Division of the Tank Automotive Research and Development Laboratory. These listings are in FORTRAN with the comments of the IFTRAN listing automatically inserted in the appropriate positions.

6.1 THE MAIN PROGRAM AND ITS STRUCTURE

The main program is called CMAIN. It allocates space for blank common and calls CMPLAR, which does all the work of the program. The subroutine CMPLAR calls subroutines according to the structure indicated in Figure 7. The structure in Figure 7 indicates which subroutines are called by the various subroutines. Figure 7 is not a subroutine tree as such. Figure 8 shows the tree structure of the program. Utility routines have been left out in both of these figures in order to simplify the description; a list of them appear below.

XMIT - initializes arrays
DOT - performs a dot product
XNORM - takes the norm of a vector
CODE - changes region I.D. from form used in program to more readable form for output
TIME - gets computer time

The commented IFTRAN listing of the main subroutine appears in Figure 9. It spells out in detail how the program works, from input, through optimization, to output of region statistics.

6.2 INPUT ROUTINES: SMNPUT AND SMINDAT

The two input routines are SMNPUT and SMINDAT, which are called in that order by the main routine, CMPLAR. Their functions can be readily determined by examination of the commented IFTRAN listings of these subroutines in Figure 10.

6.3 OPTIMIZATION ROUTINES

6.3.1 INITA

The initialization of the P-functions occurs in the subroutine INITA. The equations for this initialization are described in Section 3.2.1.

I. Input
   A. SMNPUT - processes 4 parameter lists (except DDPARM)
      1. DATADF - processes DDPARM parameter list and defines INPUT file
      2. CORMAT - computes correlation matrix

II. Optimization
   A. Perform numerical optimization
      1. INITA - initializes P-functions
      2. OPTW - optimizes weights for given hyperplane coefficients
         a. GAUUSJ - inverts G^TG matrix
      3. OPT - optimizes hyperplane coefficients and weights using gradient search
         a. Function subroutine F - called every iteration
            1. CFCN - computes RMS error, activity of hyperplanes, and gradient vector
         b. STOPG - tests stopping criteria
         c. STPOUT - output routine which displays P-functions and other data
   B. Output P-functions
      1. WSFUNC
      2. WSFUNF
   C. Compute RMS error and percent variation explained
      1. ENERGY (calls function subroutine F which calls CFCN)

III. FSTATS - Analysis of Fit by Regions
   A. MNCOV - computes overall statistics
   B. NETHYP - manipulates pointers and data in order to perform region statistics
      1. HSORT2 - sorts arrays
      2. HSORT3 - sorts arrays
      3. RGSTAT - computes region statistics
      4. ERCALC - computes region error statistics

Figure 7. Main Program and its Structure

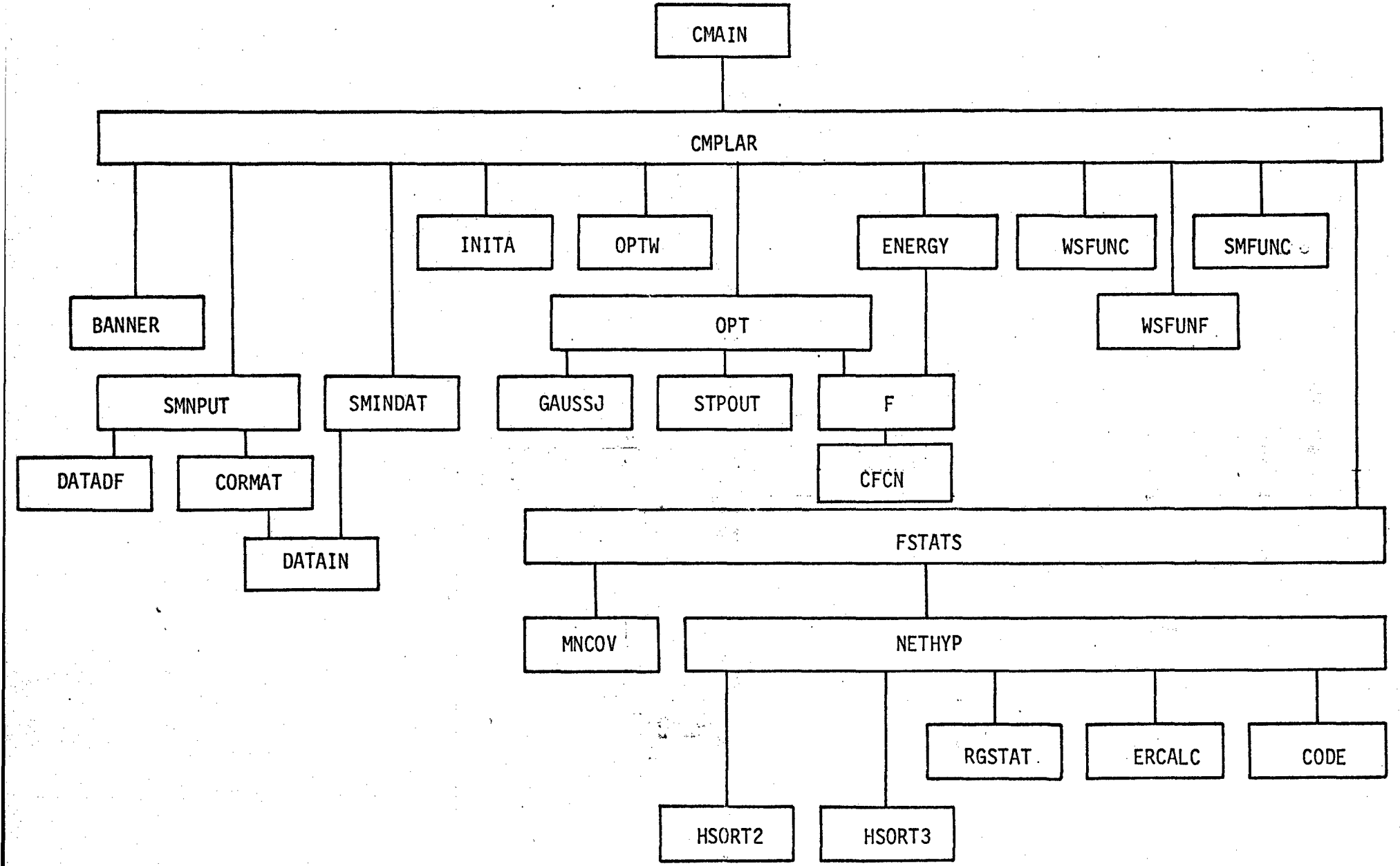


Figure 8. Tree Structure of the Program.

```
SUBROUTINE CMPLAR(AAA,LAAA)
INPUT DATA AND CONTROL PARAMETERS
IF P-FUNCTIONS ARE BEING INPUT BY USER
.   IF P-FUNCTIONS ARE IN BINARY FORMAT
.   .   READ NO. OF DIMENSIONS, NO. OF HYPERPLANES, NO. OF P-FUNCTIONS,
.   .    AND NO. OF HYPERPLANES IN EACH P-FUNCTION
.   .   IF INPUT DIMENSION DOES NOT MATCH DATA DIMENSION
.   .   .   PRINT ERROR MESSAGE AND STOP
.   .   END IF
.   .   RE-ADJUST POINTERS TO BLANK COMMON AREAS FOR INPUT P-FUNCTIONS
.   .   IF NOT ENOUGH MEMORY ALLOCATED FOR BLANK COMMON
.   .   .   PRINT ERROR MESSAGE AND STOP
.   .   END IF
.   .   READ HYPERPLANES AND WEIGHTS
.   END IF
.   PRINT INITIAL P-FUNCTIONS BEFORE STANDARDIZATION
.   INPUT DATA, STANDARDIZE DATA
.   INITIALIZE TEMPORARY DEPENDENT VARIABLE ARRAY WITH DEPENDENT
.    VARIABLE VALUES
.   INITIALIZE ACTIVITY AT HIGH LEVEL
.   IF SOME OPTIMIZATION IS TO BE DONE
.   .   STANDARDIZE P-FUNCTIONS
.   .   PRINT INITIAL STANDARDIZED P-FUNCTIONS
.   .   OPTIMIZE WEIGHTS
.   .   CALCULATE AND PRINT RMS ERROR AND PCTVE
.   END IF
ELSE - TIME HAS BEEN ALLOTTED FOR MARGINAL OPTIMIZATION,
 P-FUNCTIONS TO BE INITIALIZED BY THE PROGRAM
.   INPUT DATA, STANDARDIZE DATA
.   SET UP TIME LIMIT PARAMATERS AND PRINT CURRENT TIME
.   PRINT MESSAGE INDICATING BEGINNING OF STAGE 1 OF OPTIMIZATION
.   INITIALIZE RESIDUAL ARRAY WITH DEPENDENT VARIABLE VALUES
.   PRINT ITERATION PARAMETER LIMITS FOR STAGE 1
.   DO FOR EACH P-FUNCTION
.   .   SET UP SIZES OF ITERATION ARRAYS
.   .   INITIALIZE CURRENT P-FUNCTION USING RESIDUAL ARRAY FOR DATA
.   .   INITIALIZE WEIGHTS FOR INITIAL HYPERPLANES USING RESIDUAL ARRAY
.   .    AS DATA
.   .   CALCULATE RMS ERROR, PCTVE AND ACTIVITY
.   .   PRINT INITIAL P-FUNCTION BEING INITIALIZED
.   .   PRINT INITIAL ERROR AND PCTVE
.   .   SET UP SIZE OF TOTAL STATE VECTOR TO BE OPTIMIZED (2 IS FOR THE
.   .    WEIGHTS)
.   .   PERFORM GRADIENT SEARCH OVER CURRENT P-FUNCTION AND THE 2 WEIGHTS
.   .   CALCULATE AND PRINT RMS ERROR, PCTVE, AND ACTIVITY
.   .   PRINT P-FUNCTION BEING OPTIMIZED
.   .   OPTIMIZE WEIGHTS FOR EXISTING P-FUNCTIONS
.   .   CALCULATE AND PRINT RMS ERROR AND PCTVE
.   .   DO FOR EVERY SAMPLE POINT
.   .   .   STORE RESIDUAL OF DEPENDENT DATA LESS THE EVALUATION OF
.   .   .    EXISTING P-FUNCTIONS INTO RESIDUAL ARRAY
```

Figure 9. Commented IFTRAN Listing of Main Program (CMPLAR)

(continued on next page)

```
.    .    END DO
.    END DO
.    RESTORE VARIABLES
END IF  *    END OF STAGE 1 OF OPTIMIZATION
IF SOME TIME IS ALLOWED FOR SEQUENTIAL OPTIMIZATION
.    PRINT ITERATION PARAMETER LIMITS FOR STAGE 2
.    PRINT P-FUNCTIONS AT START SEQUENTIAL OPTIMIZATION
.    REPEAT
.    .    REPEAT
.    .    .    SET UP FIELD LENGTHS
.    .    .    SAVE THE TWO HYPERPLANE COEFFICIENTS WHICH ARE TEMPORARILY
.    .    .     OVER-WRITTEN WITH 2 WEIGHTS
.    .    .    SET WEIGHT OF CURRENT P-FUNCTION BEING OPTIMIZED TO ZERO
.    .    .    DO FOR EACH POINT IN TEMP, DEP, VAR, ARRAY
.    .    .    .    COMPUTE RESIDUAL OF DEPENDENT DATA LESS THE EVALUATION OF THE
.    .    .    .     CURRENT APPROXIMATING FUNCTION (WITH CURRENTLY OPTIMIZED P-
.    .    .    .     FUNCTION ZEROED OUT)
.    .    .    END DO
.    .    .    SET UP INDEX LIMITS FOR OPTIMIZATION
.    .    .    INITIALIZE WEIGHTS FOR RESIDUAL ARRAY
.    .    .    PERFORM GRADIENT SEARCH FOR COEFFICIENTS OF CURRENT P-FUNCTION
.    .    .     BEING OPTIMIZED, ALONG WITH ITS WEIGHT AND CONSTANT WEIGHT
.    .    .    PRINT CURRENT P-FUNCTION
.    .    .    SET WEIGHT OF CURRENT P-FUNCTION AND CONSTANT WEIGHT
.    .    .    RESTORE VARIABLES AND INCREMENT COUNTERS
.    .    UNTIL ALL P-FUNCTIONS HAVE BEEN DONE ONCE OR TIME LIMIT IS REACHED
.    .    RE-INITIALIZE TEMPORARY DEPENDENT VARIABLE ARRAY
.    UNTIL TIME LIMIT IS REACHED,  THEN PRINT MESSAGE INDICATING END OF
.     SEQUENTIAL OPTIMIZATION
.    RE-OPTIMIZE WEIGHTS FOR ALL P-FUNCTIONS TOGETHER
.    CALCULATE AND PRINT RMS ERROR, PCTVE, AND ACTIVITY
END IF  *    END OF STAGE 2 OF OPTIMIZATION
IF SOME TIME IS ALLOWED FOR JOINT OPTIMIZATION
.    PRINT ITERATION PARAMETER LIMITS FOR STAGE 3
.    PRINT INITIAL P-FUNCTIONS
.    CALCULATE AND PRINT RMS ERROR AND PCTVE
.    PERFORM GRADIENT SEARCH OVER ALL P-FUNCTIONS AND WEIGHTS
.    OPTIMIZE WEIGHTS FOR ALL P-FUNCTIONS TOGETHER
.    CALCULATE AND PRINT RMS ERROR, PCTVE, AND ACTIVITY
.    PRINT FINAL P-FUNCTIONS
END IF  *    END OF STAGE 3 OF OPTIMIZATION
DO FOR ALL HYPERPLANES
.    IF ACTIVITY IS BELOW MINIMUM
.    .    REMOVE HYPERPLANE
.    END IF
END DO
DO FOR ALL INPUT SAMPLE POINTS
.    REMOVE STANDARDIZATION FROM INDEPENDENT VARIABLES IN DATA
END DO
IF HYPERPLANES ARE CURRENTLY STANDARDIZED
.    DO FOR EACH HYPERPLANE
.    .    TRANSFORM HYPERPLANE COEFFICIENTS TO THOSE FOR NON-STANDARDIZED,
.    .     I.E. INPUT, DATA
.    END DO
END IF
IF SOME OPTIMIZATION HAS BEEN PERFORMED
.    OPTIMIZE WEIGHTS FOR P-FUNCTION
END IF
CALCULATE RMS ERROR, PCTVE, AND ACTIVITY
PRINT RMS ERROR, PCTVE, ALL P-FUNCTIONS, AND ALL WEIGHTS
IF A UNIT HAS BEEN SPECIFIED FOR SAVING P-FUNCTIONS
.    SAVE P-FUNCTIONS ON UNIT ISAVE
END IF
IF FLAG INDICATES NO FURTHER ANALYSIS, STOP
DO FOR ALL SAMPLE POINTS
.    COMPUTE PREDICTED VALUE AND ERROR
.    PRINT DATA POINT - INDEPENDENT VARIABLES, DEPENDENT VARIABLE,
.     PREDICTED VALUE, AND ERROR
END DO
PERFORM DETAILED REGION STATISTICS
STOP
END
```

Figure 9. (Cont.) Commented IFTRAN Listing of Main Program (CMPLAR)

```
SUBROUTINE SMNPUT
INITIALIZE PARAMETERS AND ASSIGN DEFAULTS
READ PARAMETERS IN FOUR NAMELISTS
FILL OPTIMIZATION PARAMETERS WITH DEFAULTS IF NECESSARY
READ DATA DEFINITION CARDS
IF NPFUNC IS NEGATIVE
.   COMPUTE STATISTICS FOR INPUT DATA
.   STOP
END IF
PRINT INPARM, STPARM, ITPARM AND OTPARM VARIABLES WITH EXPLANATION
ALLOCATE BLANK COMMON
TEST SIZE OF BLANK COMMON
RETURN
END
```

a) Commented IFTRAN Listing of Input Subroutines (SMNPUT)

```
SUBROUTINE SMNDAT
READ DATA
FIND MEAN AND STANDARD DEVIATION OF SAMPLES
NORMALIZE SAMPLES
RETURN
END
```

b) Commented IFTRAN Listing of Input Subroutines (SMNDAT)

Figure 10.

6.3.2 OPTW

The optimization of the weights occurs in OPTW. The entire function of this subroutine is to calculate the matrix $\underline{G}^T\underline{G}$ so that it can be inverted by GAUSSJ and thus produce the optimized weights. Since $\underline{G}^T\underline{G}$ is symmetric, only the upper triangular part is calculated and then the lower triangular part is filled in, one column at a time starting from the first. The error message "OPTW FAILS, MATRIX SINGULAR" is printed if the subroutine GAUSSJ passes a value of 0.0 for DET, the determinant of $\underline{G}^T\underline{G}$.

GAUSSJ calculates the inverse of the matrix $\underline{G}^T\underline{G}$ by using Gaussian elimination. It does not explicitly calculate the determinant but sets the variable DET = 0.0 when it cannot find a pivot element which is non-zero.

6.3.3 OPT

The subroutine OPT performs a gradient search over the hyperplane coefficients and weights which are being optimized in one of the three types of optimization. Figure 11 is a commented IFTRAN listing of the subroutine OPT which explains how it works.

6.3.4 The Function Subroutine F and the Subroutine CFCN

The function subroutine F simply sets up index parameters, increments the iteration counter, N_F, used in the overall iteration scheme, and calls the subroutine CFCN which gives the value of the RMS error as F.

The subroutine CFCN is only called from the function subroutine F and it performs the following function:

```
SUBROUTINE OPT(X,XTEST,DX,NXX,JJ)
DESCRIBE CALLING PARAMETERS
SET UP ARRAYS AND COMMON AREAS
OBTAIN INITIAL VALUE OF ERROR AND GRADIENT
INITIALIZE ITERATION COUNTER, SUCCESSES COUNTER, CONSECUTIVE
 SUCCESSES COUNTER, AND ERROR CHANGE
LOOP UNTIL ONE OF THE ITERATION TERMINATION CRITERIA IS SATISFIED
.   IF COMPUTING TIME EXCEEDS ALOTTED TIME
.   .   PRINT STOPPING CRITERIA AND TIME
EXIT
.   END IF
.   IF STOPPING CRITERIA OF GRADIENT SIZE, ERROR SIZE, CHANGE IN
.    ERROR, OR STEP SIZE IS MET
.   .   PRINT STOPPING CRITERIA AND TIME
EXIT
.   END IF
.   IF ITERATION COUNTER IS MULTIPLE OF PRINT FREQUENCY COUNTER
.   .   PRINT STATE VARIABLE AND GRADIENT
.   .   PRINT STOPPING CRITERIA
.   END IF
.   DO ELEMENT-WISE IN STATE VECTOR
.   .   DEFINE TRIAL STATE VARIABLE AS STATE VARIABLE MINUS THE
.   .    PRODUCT OF CURRENT STEP SIZE AND UNIT GRADIENT
.   END DO
.   EVALUATE ERROR AND GRADIENT WITH TRIAL STATE VARIABLE
.   IF ERROR WITH TRIAL STATE VARIABLE IS GREATER THAN ERROR WIT.
.    CURRENT STATE VARIABLE
.   .   DIVIDE CURRENT STEP SIZE BY FACTOR
.   .   RESTORE GRADIENT CORRESPONDING TO CURRENT STATE VARIABLE
.   .   RE-INITIALIZE CONSECUTIVE SUCCESSES COUNTER
.   ELSE TRIAL STATE VARIABLE PRODUCES SMALLER ERROR THAN CURRENT
.    STATE VARIABLE
.   .   SUBSTITUTE TRIAL STATE VARIABLE INTO CURRENT STATE VARIABLE
.   .   SUBSTITUTE TRIAL STATE ERROR INTO CURRENT STATE ERROR
.   .   INCREMENT SUCCESS AND CONSECUTIVE SUCCESSES COUNTERS
.   .   IF NUMBER OF CONSECUTIVE SUCCESSES IS SUFFICIENT
.   .   .   MULTIPLY STEP SIZE BY FACTOR
.   .   END IF
.   END IF
END LOOP
END
```

Figure 11. Commented IFTRAN Listing of OPT

1. For each sample point it evaluates each P-function, and in the course of doing so, determines which hyperplane is evaluated for each P-function
2. Determines the activity vector depending on the result of step 1
3. Calculates the gradient of the hyperplane coefficients and weights being optimized

6.3.5 STPOUT

The subroutine STPOUT is an output subroutine. It first outputs one line of information as shown in the listing. It then outputs the P-function being optimized and its gradient. The logic controlling which prints occur for which values of IPRINT and the calling parameter JJ can best be seen from the IFTRAN listing.

6.3.6 STOPG

When STOPG is called from OPT, it tests each of the stopping criteria against their limits and causes optimization to stop if any of the four stopping criteria RMS error, error change, step size, or gradient magnitude is exceeded.

6.3.7 Output P-Functions: WSFUNC and WSFUNF

The two subroutines WSFUNC and WSFUNF output the P-functions. In WSFUNC, the weights and the hyperplane coefficients are output separately. In WSFUNF, the magnitude of the weights are multiplied into the hyperplane coefficients, and the algebraic sign of the weights is separately supplied.

6.3.8 ENERGY

The subroutine ENERGY is called from CMPLAR and is used to determine the RMS error and percent variation explained, both biased and unbiased.

It does this by calling the function subroutine F (see Section 6.3.4) and implementing the definitions of RMS error and PCTVE.

6.4 FSTATS

The subroutine FSTATS calls MNCOV, prints out results of the correlation coefficient and other variables calculated in MNCOV, and then calls NETHYP.

6.4.1 MNCOV

The subroutine MNCOV computes overall statistics over all the data points. The following are calculated for all n input variables, the output variable, and the predicted variable:

1. The means and standard deviations of each variable
2. The covariance and correlation matrices for all n + 2 variables together.

6.4.2 NETHYP

The subroutine NETHYP does the following:

1. For each sample point, it computes the active hyperplanes, sets up various pointers and tables for sorts
2. It sorts a table of pointers to the data itself and another table of counters into regions of increasing numbers of points
3. Calculates statistics for each region and stores them on a scratch file
4. Reads back the contents of the scratch file for various output tables.

This subroutine can be readily understood by examining the IFTRAN listings. The comments are intended to show how the tables IWORK and INET are defined as the program progresses.

6.4.3 Statistical Subroutines Called by NETHYP: RGSTAT and ERCALC

The subroutine RGSTAT calculates the minimum, maximum, mean, and standard deviation for all n input variables, the output variable, and the predicted variable for each region.

The subroutine ERCALC calculates the RMS error, the maximum error, and the percent variation explained in each region.

APPENDIX A

APPENDIX A

SUMMARY OF NOTATIONS

COMPLIAR (COntinuous Multivariate Piecewise-Linear Approximation/Regression)

Following is a list of notations used in this document. When appropriate, the corresponding symbol used in the program is indicated in brackets.

$\underline{x} = (x_1 \ldots x_k \ldots x_n)$ vector input variable

$\underline{x}' = (x_1 \ldots x_k \ldots x_n 1)$ input variable vector with 1 augmented [AAA(IY)]

y = scalar output variable [AAA(IF)]

n = number of input variables [NDIM]

M = number of data points [NSAMP]

ℓ = index of data points ($\ell = 1,2,\ldots,M$)

$\underline{z}^{\ell} = (\underline{x}^{\ell}, y^{\ell})$ = one of the input/output data points

N_p = number of P-functions in overall fit [NPFUNC]

i = index of P-function ($i = 1,2,\ldots N_p$)

Q_i = number of hyperplanes in i^{th} P-function [NHYP]

q = index of hyperplane

$H_q^{(i)}(\underline{x})$ = q^{th} hyperplane of i^{th} P-function

$$\underline{A} = \begin{bmatrix} \underline{A}_1 \\ \vdots \\ \underline{A}_i \\ \vdots \\ \underline{A}_{N_p} \end{bmatrix}$$

overall matrix of hyperplane coefficients for the N_p P-functions [AAA(IHYP)]

$$\underline{A}_i = \begin{matrix} \\ Q_i \\ \\ \end{matrix} \begin{bmatrix} \underline{a}_1^{(i)} \\ \vdots \\ \underline{a}_q^{(i)} \\ \vdots \\ \underline{a}_{Q_i}^{(i)} \end{bmatrix}$$

matrix of hyperplane coefficients for the i^{th} P-function

$$\underline{a}_q^{(i)} = \left[a_{q1}^{(i)} \ldots a_{qk}^{(i)} \ldots a_{q,(n+1)}^{(i)}\right]$$

hyperplane coefficients for the q^{th} hyperplane of the i^{th} P-function

$|\Delta\underline{\gamma}|$ = step size in Gradient Search [TK]

$|\Delta\gamma_0|$ = initial step size (supplied by user) [input as TK, stored as OTK]

k_γ = step size adjustment factor [TKFACT]

$$P_i(\underline{x};\underline{A}_i) = \underset{1\le q\le Q_i}{\text{Max}} \left\{H_q^{(i)}(\underline{x})\right\} = i^{th} \text{ P-function}$$

$\underline{W} = [w_1, w_2, \ldots w_{N_p+1}]$ = P-function weight vector [AAA(IW)]

$$F(\underline{x}^\ell;\underline{A},\underline{W}) = \sum_{i=1}^{N_p} w_i P_i(\underline{x}^\ell;\underline{A}_i) + w_{N_p+1} = \text{general form of COMPLIAR fit}$$

$$E = \text{MSE of approximation} = \frac{1}{M}\sum_{\ell=1}^{M} [y^\ell - F(\underline{x}^\ell;\underline{A},\underline{W})]^2$$

[function subroutine F]

N_F = total number of degrees of freedom for the fit

$$g_{\ell i} = P_i(\underline{x}^{\ell}; \underline{A}_i) \qquad ; i=1,\ldots,N_p \quad \ell=1,\ldots,M$$

$$\underline{G} = \begin{bmatrix} & & & 1 \\ & g_{\ell_i} & & 1 \\ & & & \vdots \\ & & & 1 \end{bmatrix} \quad \begin{matrix} \\ M \\ \\ \end{matrix} \quad \overbrace{\qquad\qquad}^{N_{p+1}}$$

$$\underline{F} = \underline{W}\,\underline{G}^T$$

Matrix Representation of $\{F(\underline{x}^{\ell}; \underline{A}, \underline{W}); \ell=1,\ldots,M\}$

DISTRIBUTION LIST

DISTRIBUTION LIST
(As of 1 January 1977)

Please notify USATARADCOM, DRDTA-RHMM, Warren, Michigan 48090, of corrections and/or changes in address.

No. of Copies	
50	Commander US Army Tank Automotive Research and Development Command Warren, MI 48090
2	Hqs, DA Ofc of Dep Chief for Rsch, Dev & Acquisition ATTN: DAMA-CSS-D LTC Bertel R. Bertils Washington, D.C. 20310
1	Superintendent US Military Academy ATTN: Dept of Engineering Course Director for Automotive Engineering West Point, NY 10996
1	Commander US Army Logistic Center ATTN: ATCL-CC Mr. J. McClure Ft. Lee, VA 23801
1	US Army Research Office P.O. Box 12211 ATTN: Mr. James Murray Research Triangle Park, NC 27709
1	Hq, DA ATTN: DAMA-ARZ-D Dr. Herschner Washington, D.C. 20310
1	Hq, DA Ofc of Dep Chief of Staff for Rsch, Dev & Acquisition ATTN: DAMA-REZ-E Dr. Charles Church Washington, D.C. 20310

1 Commander
US Army Concept Analysis Agency
Long Range Studies
8120 Woodmont Avenue
Bethesda, MD 20014

1 Dept of the Army
Office Chief of Engineers
Chief Military Programs Team
Rsch & Dev Office
ATTN: DAEM-RDM
Washington, D.C. 20314

2 Director
US Army corps of Engineers
Waterways Experiment Station
P.O. Box 631
Vicksburg, MS 39180

3 Director
US Army Corps of Engineers
Waterways Experiment Station
ATTN: Mobility & Environmental Laboratory
P.O. Box 631
Vicksburg, MS 39180

4 Director
US Army Cold Regions Research & Engineering Lab
ATTN: Dr. Freitag, Dr. W. Harrison, Dr. Liston,
Library
P.O. Box 282
Hanover, NH 03755

2 President
Army Armor and Engineer Board
Ft. Knox, KY 40121

1 Commander
US Army Arctic Test Center
APO 409
Seattle, Washington 98733

2 Commander
US Army Test & Evaluation Command
ATTN: AMSTE-BB and AMSTE-TA
Aberdeen Proving Ground, MD 21005

2 Commander
Rock Island Arsenal
ATTN: SARRI-LR
Rock Island, IL 61201

2	Commander US Army Yuma Proving Ground ATTN: STEYP-RPT, STEYP-TE Yuma, Arizona 85364
1	Mr. Frank S. Mendez, P.E. Technical Director US Army Tropic Test Center ATTN: STETC-TA Box 942 Fort Clayton, Canal Zone 09827
1	Commander US Army Natick Laboratories ATTN: Technical Library Natick, Massachusetts 01760
1	Director US Army Human Engineering Laboratory ATTN: Mr. Eckels Aberdeen Proving Ground, MD 21005
2	Director US Army Ballistic Research Laboratory Aberdeen Proving Ground, MD 21005
3	Director US Army Materiel Systems Analysis Agency ATTN: AMXSY-CM, Messrs. D. Woomert, W. Niemeyer, W. Criswell Aberdeen Proving Ground, MD 21005
12	Director Defense Documentation Center Cameron Station Alexandria, VA 22314
1	US Marine Corps Mobility & Logistics Division Development and Ed Command ATTN: Mr. Hickson Quantico, VA 22134
1	Mr. A. M. Wooley West Coast Test Branch Mobility and Support Division Marine Corps Base Camp Pendleton, CA 92055

1 Naval Ship Research & Dev Center
Aviation & Surface Effects Dept.
Code 161
ATTN: E. O'Neal & W. Zeitfuss
Washington, D.C. 20034

1 Director
National Tillage Machinery Laboratory
Box 792
Auburn, Alabama 36830

1 Director
USDA Forest Service Equipment Dev Center
444 East Bonita Avenue
San Dimes, CA 91773

1 Director
Keweenaw Research Center
Michigan Technological University
Houghton, MI 49931

1 Engineering Societies Library
345 East 47th Street
New York, NY 10017

1 Dr. M. G. Bekker
224 East Islay Drive
Santa Barbara, CA 93101

1 Dr. I. R. Ehrlich, Dean for Research
Stevens Institute of Technology
Castle Point Station
Hoboken, NJ 07030

2 Grumman Aerospace Corporation
ATTN: Dr. L. Karafiath, Mr. E. Markow
Plant 35
Bethpage, Long Island, NY 11714

1 Dr. Bruce Liljedahl
Agricultural Engineering Department
Purdue University
Lafayette, IN 46207

1 Dr. W. G. Baker
Dean, College of Engineering
University of Detroit
4001 W. McNichols
Detroit, MI 48221

1 Mr. H. C. Hodges
Nevada Automotive Test Center
Box 234
Carson City, NV 89701

1 Mr. W. S. Hodges
Lockheed Missile & Space Corporation
Ground Vehicle Systems
Department 50-24, bldg 528
Sunnyvale, CA 94088

1 Mr. R. D. Wismer
Deere & Company
Engineering Research
3300 River Drive
Moline, IL 61265

1 Oregon State University
Library
Corvallis, Oregon 97331

1 Southwest Research Institute
ATTN: Mr. R. C. Hemior
8500 Culebra Road
San Antonio, TX 78228

1 FMC Corporation
Technical Library
P.O. Box 1201
San Jose, CA 95108

1 Mr. J. Appelblatt
Director of Engineering
Cadillac Gauge Co.
P.O. Box 1027
Warren, MI 48090

2 Chrysler Corporation
Mobility Research Laboratory,
Defense Engineering
ATTN: Dr. B. VanDeusen, Mr. J. Cohron
Department 6100
P.O. Box 751
Detroit, MI 48231

1 Library
CALSPAN Corporation
Box 235
4455 Genesse Street
Buffalo, NY 14221

1 Mr. Sven E. Lind
SFM, Forsvaretsforskningsanstalt
Avd 2
Stockholm 80, Sweden

2 Mr. Rolf Schreiber
c/o Bundesamt Fuer Wehrtechnik
Und Beschaffung - KGII 7 -
5400 Koblenz
Am Rhein 2-6 Germany

1 Foreign Science & Technology Ctr
220 7th Street North East
ATTN: AMXST-GEI
Mr. Tim Nix
Charlottesville, VA 22901

1 General Research Corporation
ATTN: Mr. A. Viilu
7655 Old Springhouse Road
Westgate Research Park
Mc Lean, VA 22101

1 Commander
US Army Development & Readiness Command
ATTN: Dr. Norm Klein
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Battelle
Columbus Division
Advanced Concepts Laboratory
P.O. Box 339
Warren, Michigan 48090

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-12242	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* REPRO-MODELING APPLIED TO THE SIMPLIFICATION OF TARADCOM COMPUTER MODELS		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER TSC-PD-A156-1
7. AUTHOR(s) Gene B. Goldstein Theodore Dushane		8. CONTRACT OR GRANT NUMBER(s) DAAE07-76-C-0105
9. PERFORMING ORGANIZATION NAME AND ADDRESS Technology Service Corporation 2811 Wilshire Blvd Santa Monica, California 90403		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601A 1F662601AH91
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Tank Automotive Research & Development Command, TARAD Labs, Science & Technology Division, DRDTA-RH, Warren, MI 48090		12. REPORT DATE December 1976 13. NUMBER OF PAGES 128
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Mathematical Modeling
Mathematical Modeling Techniques
Repromodeling
Continuous Piecewise Linear Regression

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

As mathematical models become large and complex, they become more difficult to apply and understand. Repro-modeling is a technique for constructing an efficient input/output approximation to a complex model. The technique is founded upon an economical representation for continuous piecewise linear functions of many input variables. Repro-modeling as described here is the fitting of input/output data by a continuous piecewise linear function. A software package called COMPLIAR which accomplishes this is described, and the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

theory underlying its use is explained. Instructions for the use of the software and examples are included.